NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SIMULATING MISSION COMMAND FOR PLANNING AND ANALYSIS

by

Hasan Beker

June 2015

Thesis Advisor:	Arnold Buss
Second Reader:	Leroy A. Jackson

This thesis was performed at the MOVES Institute

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2015	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE SIMULATING MISSION COMMAND FOR PLANNING AND ANALYSIS			**5. FUNDING NUMBERS**
6. AUTHOR(S) Hasan Beker			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Time is of importance for the success of a military mission and is a beneficial tool that can provide an advantage over the enemy if used effectively. This thesis prototypes a mission planning and analysis simulation for commanders and staff officers to assist in the mission-planning process and helps them to manage the duration of a mission by using technology and mission command as a means to accelerate mission. The simulation and analysis of the output data in this thesis has five main benefits. Firstly, it finds the critical path of the task network system of a mission planned according to the mission command concept. Secondly, it gives the estimated success rate of finishing the mission on time. Thirdly, it provides an interactive mission-planning process and amortizes most critical factors of war such as morale, leadership, weather, and terrain. Fourthly, it saves time and money spent for the mission by focusing attention on only a small part of the tasks planned during the mission-planning process. Lastly, the Excel sheets developed in the course of this work can be used as a matrix for synchronizing the mission plan.

14. SUBJECT TERMS Mission Planning, CPM, PERT, Simulation, DES, Simkit, Triangle Distribution, Critical Path			**15. NUMBER OF PAGES** 121
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

SIMULATING MISSION COMMAND FOR PLANNING AND ANALYSIS

Hasan Beker
First Lieutenant, Turkish Army
B.S., Turkish Military Academy, 2006

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN
MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION (MOVES)**

from the

**NAVAL POSTGRADUATE SCHOOL
June 2015**

Author: Hasan Beker

Approved by: Arnold Buss
Thesis Advisor

Leroy A. Jackson
Second Reader

Christian Darken
Chair, MOVES Academic Committee

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Time is of importance for the success of a military mission and is a beneficial tool that can provide an advantage over the enemy if used effectively. This thesis prototypes a mission planning and analysis simulation for commanders and staff officers to assist in the mission-planning process and helps them to manage the duration of a mission by using technology and mission command as a means to accelerate mission. The simulation and analysis of the output data in this thesis has five main benefits. Firstly, it finds the critical path of the task network system of a mission planned according to the mission command concept. Secondly, it gives the estimated success rate of finishing the mission on time. Thirdly, it provides an interactive mission-planning process and amortizes most critical factors of war such as morale, leadership, weather, and terrain. Fourthly, it saves time and money spent for the mission by focusing attention on only a small part of the tasks planned during the mission-planning process. Lastly, the Excel sheets developed in the course of this work can be used as a matrix for synchronizing the mission plan.

THIS PAGE INTENTIONALLY LEFT BLANK

DISCLAIMER

The reader is cautioned that the computer program developed in this thesis may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the planner.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ADP	Army Doctrine Publication
ADRP	Army Doctrine Reference Publication
AO	Area of Operation
AK-47	Automate Kalashnikov 1947
BCT	Brigade Combat Team
BDE	Brigade
BNTF	Battalion Task Force
CO	Company
CPM	Critical Path Method
DES	Discrete Event Simulation
FA BAT	Field Artillery Battalion
FEL	Future Event List
FIST	Fire Support Team
GPS	Global Positioning System
G3	Operations, Plans and Training Staff at Corps and Brigades
HNG	Home Nation Government
HQ	Headquarters
JMP	Statistical Software
MC	Mission Command
PERT	Program Evaluation and Review Technique
R	Statistical Programming Language
RPG-7	Rocket Propelled Grenade
UGS	Unmanned Ground System
UAS	Unmanned Aircraft System

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my thesis advisor Professor Arnold Buss and second advisor Leroy Jackson, for their guidance, support, and patience throughout the thesis writing process.

I also would like to thank my beloved wife, Yasemin, for her support throughout my entire study at NPS.

THIS PAGE INTENTIONALLY LEFT BLANK

I. BACKGROUND

This objective of this thesis is to provide a mission planning and analysis program for commanders and staff officers in their mission planning process by simulating mission command and helps them to see the acceleration effect of technology and mission command on the overall duration of missions. To be able to understand some critical tools used to design the simulation explained in this thesis and give a general picture of the structure of the thesis, Mission Command Concept, Discrete Event Simulation, Simkit Application and Task Networks will be discussed respectively in this chapter.

A. MISSION COMMAND

1. What is Mission Command?

The history of war is as old as the history of humans. In a war, there are armies, and these armies have commanders managing their contingents and giving orders to their subordinates in order to reach the ultimate goal of "winning the war." When we look at the commanding process, we face two fundamental approaches to command. The first one is a classical approach, *detailed command*, and the second one is *mission command*. Detailed command puts an emphasis on the task and detailed planning. Subordinates must rigorously follow the plan without changing any part of it when they get detailed command. This type of command should be utilized when executing missions that require fastidiousness and a great deal of coordination (Ben-Shalom & Shamir, 2011). On the other hand, according to Army Doctrine Publication (ADP) 6-0, *Mission Command*, mission command is defined as "the exercise of authority and direction by the commander using mission orders to enable disciplined initiative within the commander's intent to empower agile and adaptive leaders in the conduct of unified land operations" (Department of the Army, 2012a, p. 1). As understood from this definition, the main facets of mission command are mission orders, initiative, commander's intent, and agile and adaptive leaders. In the same publication, these are also stated as the principles of mission command. The six principles of mission command listed in ADP 6-0 are as follows:

- Build cohesive teams through mutual trust.
- Create shared understanding.
- Provide a clear commander's intent.
- Exercise disciplined initiative.
- Use mission orders.
- Accept prudent risk.
  (Department of the Army, 2012a)

In the mission command approach, the control of commanders over their subordinates is never questioned. In mission command, one of the main tasks of the staff undertaking the command is to adjust the degree of control over subordinates. Every mission needs a different degree of control. For instance, a commander cannot use the same amount of control in an air-landing operation as they would in an ambush carried out by ground maneuver teams (Flynn & Schrankel, 2013).

In a conflict of war, the equipment used for communication and positioning may be destroyed or may be unusable due to poor terrain and weather conditions. In such conditions, if a subordinate waits for a new command, he or she will risk losing the initiative to act to prevent a bad situation. Every soldier knows that these kinds of situations are encountered frequently in a war. The most suitable response for a warfighter is to be ready to overcome this problem every time it occurs and to know how to take risks and initiative. So, when soldiers face situations in which they are unable to communicate easily with their superiors, it is essential that the soldiers are trained according to mission command doctrine.

In mission command, a commander gives his or her subordinates clearly stated orders without explaining all the necessary details to them. The subordinate leaders are allowed a lot of initiative and freedom in the planning and executing of these orders. In order to accomplish these orders successfully and in a certain time period, subordinate leaders must understand the intent of their commanders and execute the orders under the monitoring and control of their superiors. Sometimes their execution of a special order can violate the orders received from other commanders. The most important element here is that subordinates have the ability to take risks to execute their orders and that they have the confidence to choose the best action to implement without hesitation. To be able to acquire these abilities, subordinates should have the required mission-command training so they can act independently and flexibly and use initiative in the chaos of war.

2. Difficulties in the Practice of Mission Command

Before entering a war, commanders and their staff prepare detailed plans according to the intelligence coming from various intelligence sources and issue the plans to their subordinates. However, after a while, the current orders are no longer applicable, and it becomes necessary to update them in light of emerging developments. It takes a lot of time to get new intelligence and to update plans and then issue them again. During this time, the situation may change again, so the subordinates may lose initiative and the chance to seize opportunities while awaiting new orders. Thus, the best way for commanders to handle this kind of situation is to give the initiative to subordinates who can see more of the details of the war and the enemies in their area of responsibility. Consequently, I reach a conclusion that giving mission orders consisting of the executer and aim of the order, and intent of the commander is preferable to giving detailed and complicated orders. However, is it that simple for commanders to apply the doctrine of mission command to their commanding system? Are there any restrictions that would hinder commanders from using this war doctrine as a valuable tool in wartime?

As in the U.S. Army, mission command is accepted as a military doctrine by most other countries' armies. However, due to the large number of personal, cultural, technological, and organizational restrictions of these armies' military applications, putting the mission command doctrine into practice is difficult in spite of its successful implementations throughout history (Ben-Shalom & Shamir, 2011). The Gulf War provides some examples of these kinds of cultural and organizational restrictions. The Gulf War is considered to be a great success by most military experts. However, the ultimate goal of this war—ensuring security and stability of the Gulf region—was not achieved. Thus, after a decade, the U.S. Army had to face the Iraqi army again. In this war, even though mission command had been accepted as a war doctrine at that time, the commanders of allied armies' used detailed plans to micromanage their armies. According to some critics, if coalition forces had succeeded in implementing mission command in the Gulf War, it would have been possible to terminate the Saddam regime completely (Pryer, 2013). Another deterrent to the adoption of mission command is developing technology. It has become easy for commanders to manage their contingents through high-tech communication devices; they are able to see detailed movement of their warfighters on the war theater via GPS and monitoring systems. As early as the Vietnam War, commanders gave orders to their contingents by shouting down from helicopters (Shamir, 2011). The personal impediment that prevents commanders from using mission command is the human tendency to hate risk. This tendency causes commanders to tightly control their subordinates' every movement in a fight. Especially in modern times, the media also has a large impact on this risk-avoidance tendency because every phase of war is publicized. Because of the public's intolerance for casualties, leaders avoid taking risks and avoid giving the ultimate initiative to their subordinates, as well (Ben-Shalom & Shamir, 2011).

3. History of Mission Command

After Napoleon Bonaparte's defeat of the Prussian Army at the twin battles of Jena and Auerstedt in 1806, the Prussian Army started a fundamental reform under the leadership of the chief of the Prussian General Staff, Gerhard von Scharnhorst (Pryer, 2013). He took lessons from this defeat and started to build his army from the beginning by placing the highest priority on the education of young officers. He founded a new General Staff and Military Academy. In this academy, the main goal of the education of junior leaders was empowering them to take initiative and make independent decisions in the face of the complications of war (Shamir, 2011). As a protégé of Scharnhorst, Carl Von Clausewitz (2007) explained war as a "realm of uncertainty" in his famous book *On War*. He underlined the "uncertainty" due to the fact that it is so difficult to make the right decisions in the chaos of war. By doing so, he also emphasized indirectly the importance of mission command. Another renowned commander who contributed to the development of the doctrine of mission command is Helmuth von Moltke the Elder. He is also known as the father of mission command. He served as chief of staff for his country for 30 years and during his employment, he always placed great significance on the education and training of junior officers and non-commissioned officers; he wanted them to get enough tactical education about decentralized command. Even in their training, trainees were forced to take the utmost initiative and were required to refuse to obey their superiors' orders in some scenarios (Pryer, 2013). In the end, Prussia was able to defeat France in 1870 with soldiers trained in mission command doctrine.

In the beginning of the Second World War, Germany, as a successor of Prussia, achieved great success by using "lightning war doctrine." Decentralized command was one of the main components of this doctrine ("Blitzkrieg," n.d.). As described in this section, mission command was a war doctrine invented and mostly implemented by Prussians and their German successors. However, there are examples of mission command in U.S. Army doctrine as well. For instance, in the American Civil War, a splendid example is the order given by LTG Ulysses S. Grant to MG William T. Sherman for a campaign in 1864. In this order, LTG Grant expressed the aim of the operation without explaining the details of it by saying, "I do not propose to lay down for you a plan of Campaign, but simply to lay down the work it is desirable to have done and leave you free to execute in your own way. Submit to me however as early as you can your plan of operation" (Department of the Army, 2012b). General George S. Patton, Jr., one of the most notable characters of the Second World War, also used mission command orders and stressed their importance by voicing, “Never tell people how to do things, tell them to do, and they will surprise you with their ingenuity” (Pryer, 2013). Today, it is obvious that the U.S. Army still gives great weight to mission command and even has accepted it as a war doctrine when two modern Army Doctrine Publications are considered—ADP 6-0 and ADRP 6-0—in which the principles and details of mission command are clearly described, as well as explanations for how to implement it in modern warfare.

B. DISCRETE EVENT SIMULATION

Since simulation is a crucial tool the mission command scenario considered in this thesis, we will first give some background on discrete event simulation methodology used.

Simulations can be defined as imitations of systems during a time interval. According to their representation of time, there are two categories: time-step simulation and discrete event simulation (DES). In time-step simulation, time advances in discrete increments. On the other hand, the events in DES occur at arbitrary times, not multiples of a time step increment. It is taken for granted that there is no other change between these events, so simulation can be seen as time leaping from one event to another. According to information stated above, DES can be defined as a simulation branch that models a system in a distinct set of events during a time period by showing them at a particular moment, instead of showing them continuously (Sanchez, 2007). There are four elements of DES (Buss, 2011). These are

- States
- Events
- Parameters
- Scheduling relationships between events

1. States

A state variable is a DES component which can be changed at least once during a simulation run. The total number of available servers in a customer-queue-server system is an example of a state variable. The value of a state variable during a time period is defined as a state trajectory. A trajectory must be constant for a limited time and change suddenly to another value; in this value, it must also be stable for another certain period of time (Buss, 2011).

2. Events

Events are situations that directly affect the state in simulations by increasing or decreasing their values. Starting a service in a customer-queue-server system is an example of an event. An event should be created for every possible state transition. Besides that, every event needs an event time. Once simulation reaches the event time of an event, this event is carried out. Lastly, one of the missions of an event is scheduling another event sometime in the future (Buss, 2011).

3. Parameters

The second kind of variable after state variables is parameters. The main difference between these two variables is based on whether they change their values or not. As explained previously, state variables have the possibility of changing their values during a simulation run; however, the values of parameters do not change. A good example of parameters is the number of servers in a queuing system.

4. Next Event Algorithm

In DES systems, next event algorithm (see Figure 1) is the term for the progression of time. As mentioned before, in DES models, instead of advancing in a steady way, time is progressing in a disordered way by jumping from one event time to another. With a good understanding of next event algorithms, it is easy to create effective DES models (Buss, 2011).

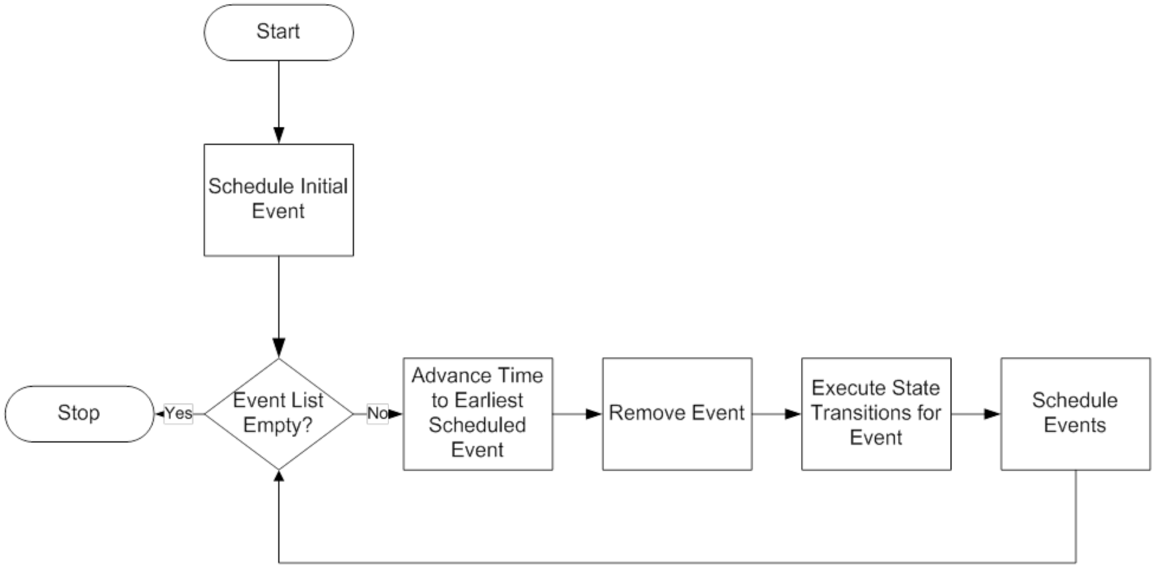


Figure 1. Next Event Algorithm (from Buss, 2011)

5. Future Event List

Events are listed and managed in a list called a future event list (FEL). This list is also responsible for organizing all information for events and keeping them in order. The content of events in an event list includes the scheduled event time and the name of current event. One of the main responsibilities of FEL is adding next events and removing the finished ones from the list. After finishing all the events in the list, FEL gets emptied and DES is stopped automatically (Buss, 2011).

Current Time	Current Event	Q	S	L	Event List
0.0	-	-	-	-	0.0 Run
0.0	Run	0	1	0	5.6 Arrival
5.6	Arrival	1	1	1	5.6 StartService 8.5 Arrival

Figure 2. Future Event List Example (from Buss, 2011)

6. Terminating Conditions

In addition to the termination of DES after an event list gets emptied, there are other ways to terminate a simulation run. The first one is using a special event “stop” that empties the event list without impacting any state. The simulation run will be ceased by this “stop” event after a certain amount of time has elapsed. The second way to terminate a simulation run is executed in FEL by counting a particular event. When FEL reaches the desired count of this special event, FEL is emptied and simulation stops. The last way to terminate is to reach a certain event that has a determined value of stopping the simulation run. Termination is easily implemented by the scheduling attribute of an event by scheduling a “stop” event when its expected value is attained (Buss, 2011).

7. Event Graphs

Event graphs are a graphical way of showing the relationship between processed events and scheduled events. Nodes and directed edges are the main elements to indicate a simulation activity. A node is a graphical presentation of an event, or a state transition, and scheduling of other events is represented by edges. It is sometimes possible to see Boolean variables or time delay on edges that construct a logical bridge between nodes. Figure 3 illustrates a basic event graph. It can be interpreted as follows: “the occurrence of Event A causes Event B to be scheduled after a time delay of t, providing condition (i) is true” (Buss, 1996, p. 2).

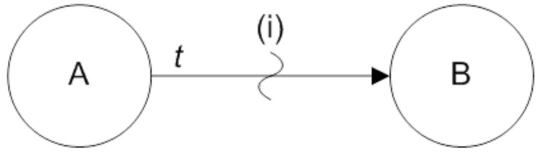


Figure 3. Basic Event Graph (from Buss, 2011)

There are two main concepts that are very useful in designing simple DES models. The first one is Scheduling Edge with Arguments and Events with Parameters, and the other one is Cancelling Edges.

Scheduling Edge is illustrated in Figure 4 and interpreted as follows: "When Event A occurs, then if condition (i) is true, Event B is scheduled to occur (placed on the Event List) after a delay of t, and when it occurs its parameter k will be set to the value of the expression j at the time it had been scheduled" (Buss, 2011).

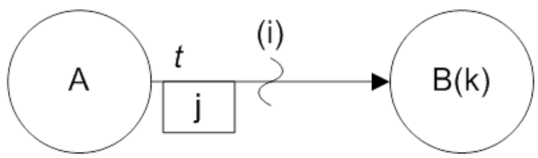


Figure 4. Scheduling Edge With Arguments and Events With Parameters (from Buss, 2011)

Cancelling an Edge is illustrated in Figure 5 and interpreted as follows:

> Whenever Event A occurs, then (following its state transition), if condition (i) is true, then the earliest scheduled occurrence of Event B is removed from the Event List. If no such Event had been previously scheduled, then nothing happens; it is not considered an error. If Event B had previously been scheduled multiple times, then only the earliest scheduled one is removed and the remaining ones are left on the Event List. (Buss, 2011)

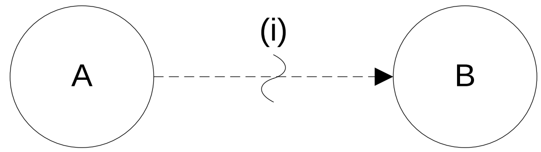


Figure 5. Cancelling an Edge (from Buss, 2011)

Another concept that makes creating event graph modeling easier is priorities on scheduling edge. In some cases, it becomes imperative to break ties for events that are scheduled simultaneously. Allocating a priority on a scheduling edge gives a preference to the following event. Figure 6 illustrates a scheduling edge with priority set to p (Buss, 2011).

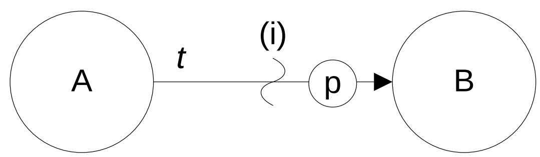


Figure 6. Scheduling Edge With Priority (from Buss, 2011)

With basic event graph component, it is easy to create effective DES models. However, when attempting to model larger DES with many events, it becomes difficult to create them. Using event graph components is a helpful way to alleviate this problem. Event graph components are modular simulation tools that have their own parameters, state variables, and events (Buss, 2011).

One of the important event graph components is SimEventListener pattern. Events in this component can impact the state of other components. SimEventListener component's working mechanism is stated as follows:

> One simulation component shows interest in another's events by explicitly being registered as a SimEventListener to it. If there is a listener relationship [as in Figure 7], then whenever an Event from Source occurs, then after it has executed its state transitions and scheduled Events, the Event is sent to Listener. If Listener has an Event that is identical (in both name and signature) to the one it "hears" then it processes that Event as if it had scheduled it. The listening component does not re-dispatch heard Events to its listeners, if it has any. (Buss, 2011)

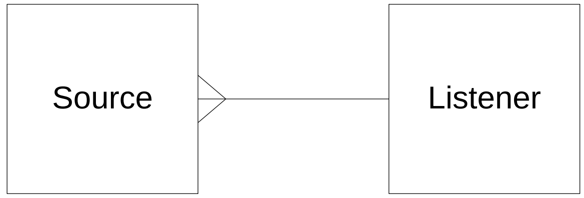


Figure 7. SimEventListener Relationship (from Buss, 2011)

The other helpful way to make larger DES models easier is the adapter pattern. It helps to trigger an event in one component by using another event of a different name in a different component: “If a ‘source’ component has an Event A and it is desired to cause Event B in a Listener component whenever A occurs, then an adapter between the Source and Listener is created that ‘adapts’ Event A to Event B” (Buss, 2011). The source event and adapted event must have the same parameter list; otherwise the model is not working properly. An adapter is illustrated in Figure 8.

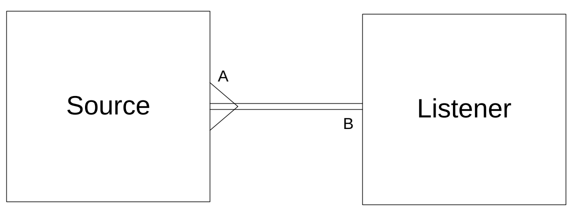


Figure 8. Prototype Adapter: Event A in Source Causes Event B in Listener (from Buss, 2011)

C. SIMKIT

Simkit is an open-source Java software package designed to create DES by implementing event graph models as Java codes. This application is designed and is still being developed by Professor Arnold Buss. Every event graph component has a counterpart in Simkit. Table 1 illustrates the relationship between the fundamental elements of event graph models and their corresponding Simkit implementations.

Table 1. Event Graph Components and Their Simkit Counterparts (Buss, 2011)

Event Graph	Simkit
Simulation Component	Subclass of SimEntityBase
Event Graph Parameter	Private instance variable, setter and getter
State Variable	Protected instance variable, getter, no setter
Event	'do' method
Scheduling Edge	Call to waitDelay() in scheduling event's 'do' method
Run Event	reset() method to initialize state variables; doRun() method to fire PropertyChange events for time-varying state variables
Event scheduled from Run event	Call to waitDelay() in doRun() method
Event scheduled from any Event	Call to waitDelay() in scheduling event's 'do' method
Event cancelled from any Event	Call to interrupt() from canceling event's 'do' method
Priority on Scheduling Edge	Priority instance as third argument to waitDelay()
Argument(s) on Events	Arguments in corresponding 'do' method
Parameter(s) on Edges	Add parameter values/expressions last (in correct order) in waitDelay()
Canceling Edge	Call to interrupt()

Every event graph model has at least one run event. The run event is represented in the event list with a time of “0.0.” It contains doRun() and reset() methods. The reset() method initializes state variables and the doRun() method is firing the corresponding PropertyChangeEvents and scheduling other events. If the model does not find a run event at the beginning of a simulation, it doesn’t start (Buss, 2011). Parameters are represented as private instance variables; on the other hand, states are represented as protected instance variables. The events are implemented by doEventName() methods and must start with a “do” string. For scheduling edges, the waitDelay() method is used; this method consists of at least two arguments: the following event name and the time delay.

As shown in Figure 3, a basic event graph model represents a scheduling edge between Event A and Event B and can be interpreted as follows: “the occurrence of Event A causes Event B to be scheduled after a time delay of t, providing condition (i) is true” (Buss, 1996).

The following snip code is the Simkit implementation of the basic event graph model in Figure 3; Figure 3 is shown again here as Figure 9.

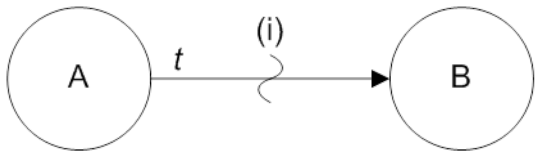


Figure 9. Basic Event Graph (from Buss, 2011)

Snip Code of Basic Event:

```
public void doA() {
        // State transitions for Event A
        if (i) {
                waitDelay(“B”, t);
        }
}
public void doB() {
        // State transitions for Event B.
}
```

(Buss, 2011)

D. TASK NETWORKS

Task networks are very efficient and useful ways to solve large and complex management problems. In military missions, planning, management, scheduling, allocation of resources and controlling are the main components to be considered very carefully to reach the desired objective. They resemble event graphs in DES while representing the events in nodes and edges in arcs. However, unlike Event Graphs, time passes when an activity node is “active”, and the edges represent precedence relationships. In Figure 10, an example of a task network diagram is shown.

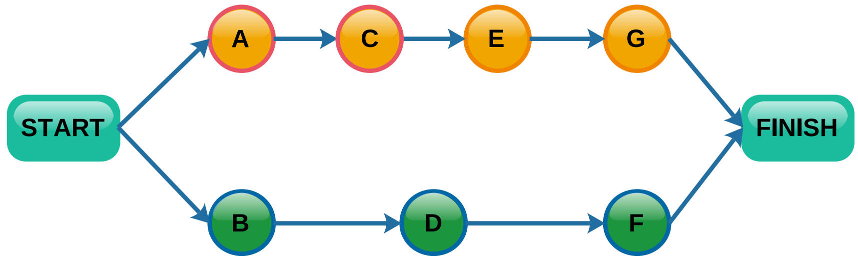


Figure 10. Task Network Diagram Example

The main goal of planning, scheduling, and controlling in a military mission is for the commander to determine the shortest and fastest way to succeed in the mission. With these tools, a commander has the ability for decision-making without needing complicated computations and exhaustive analysis (Wiest & Levy, 1969).

In this thesis, finding the critical path of the task network that represents the mission command plan is the main goal. Because of that goal, instead of naming the final product of using these methods a “project,” we call it a mission command plan. While creating a military plan, commanders and their staff list a lot of smaller tasks, and it is not easy to keep track of every detail. If a smaller task or any detail about the mission is forgotten, it will affect the achievement of the mission. By applying these methods to create a mission command plan, commanders and their staff can easily visualize every small task and activity and see the complete picture. In a network diagram of task networks, small tasks are represented as nodes and their relations as arcs.

To be able to find a critical path on a network diagram, there are two main steps to complete.

The first step is listing all tasks in a plan. For each task, the task name, task description, responsible unit or person duration, and succeeding tasks must be shown. An example of this list is illustrated in Table 2.

Table 2. Task List

Task Name	Task Description	Unit or Person Responsible	Duration	Succeeding Tasks
A	Receive Warning Order	Unit 1, 2	1 hour	B, C
B	Conducting Initial Mission Planning	Unit 1, 2 Commanders	3 hours	D
C	Occupy Positions	Unit 1, 2	5 hours	D
D	Conduct Rehearsals	Unit 1, 2	6 hours	E
E	Issue Execute Order	Unit 1, 2 Commanders	1 hour	

The second step is plotting activities in nodes and arcs. Nodes (circles) show activities within the project. The name of the task is stated in these circles, and the durations of the activities are on the nodes. The arcs (arrows) between two circles demonstrate the activities required for finishing the tasks. In Figure 11, a network diagram of the tasks listed on Table 2 is shown by nodes and arcs.

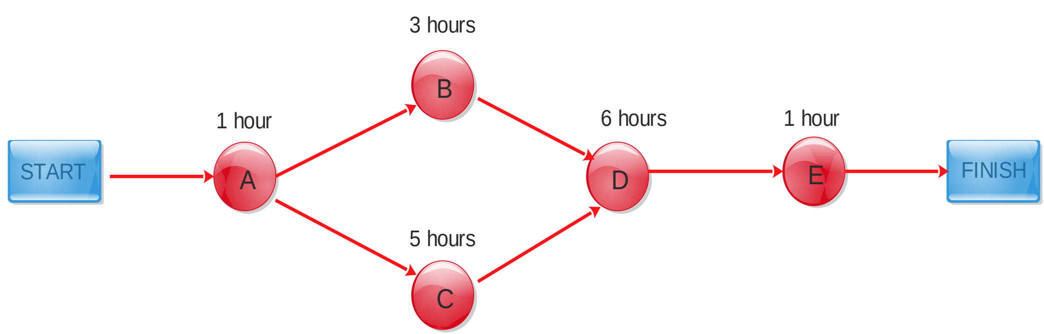


Figure 11. Task Network Diagram

The longest time from start to finish is called the critical path; it also determines the earliest project finish time. Tasks on the critical path are called critical tasks. All the tasks (nodes) must be finished before starting a new task (node). In the diagram in Figure 12, the highlighted A-C-D-E path is visible, which is the critical path for this task network example.

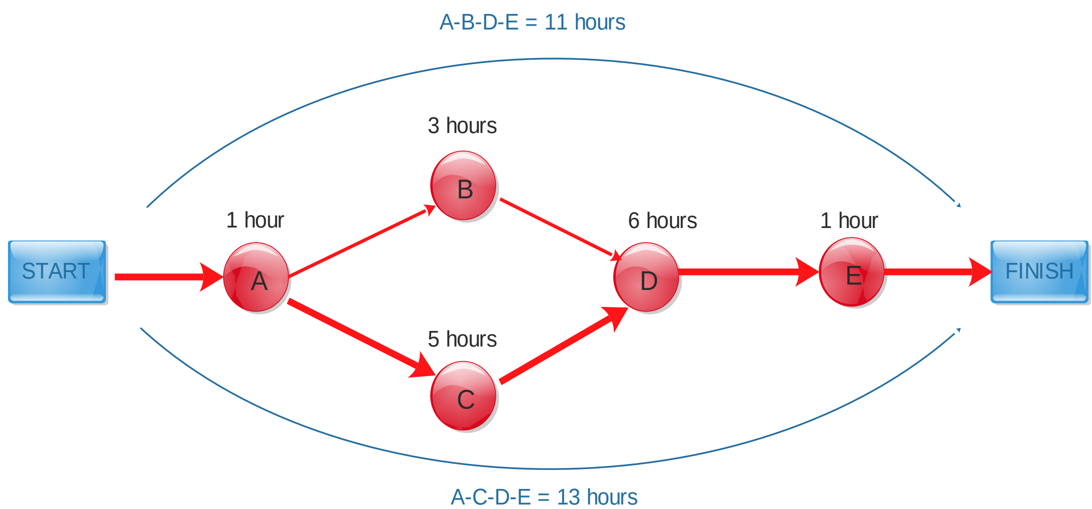


Figure 12. Critical Path on a Task Network Diagram

In Figure 12, the critical path is found by evaluating the duration of the tasks. However, in more complex situations, it is not as easy to find the critical path and total duration of the plan because of the possibility of confronting some parallel tasks whose durations are the same. Using task network boxes is an effective way to overcome this problem. To find the duration time of the complete plan and the critical path, each task must be allocated with an early start time, late start time, early finish time, late finish time, slack time, and their durations (Wiest & Levy, 1969). In Figure 13, nodes are illustrated by task boxes, which are of importance in finding the critical path.

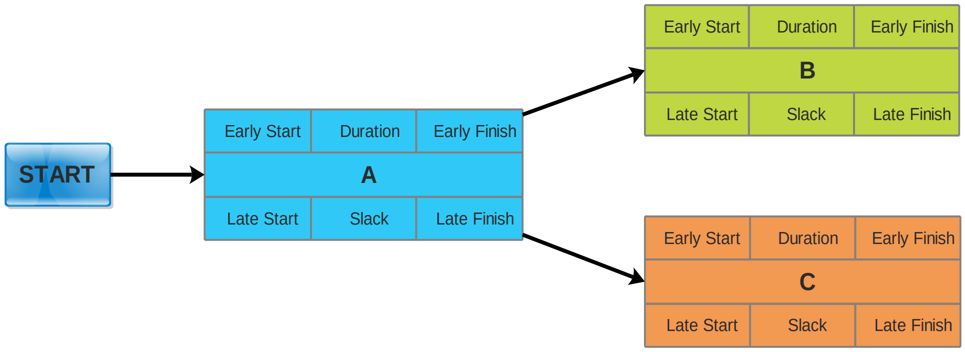
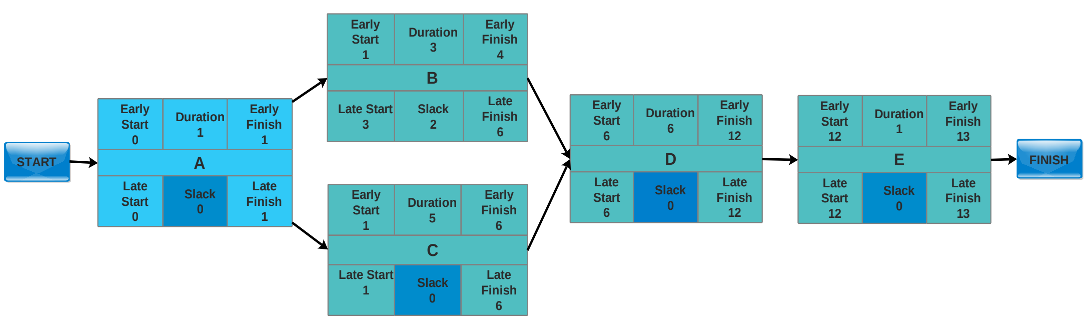


Figure 13. Illustration of Tasks in Boxes

There are some simple equations to find the slots given in Figure 13. As stated in the task list in Table 1, durations of the tasks are already given and the early start of the tasks after start is zero. Early finish time is equal to duration plus early start, and late finish time is equal to late start plus duration. Slack time, which is explained in detail in the next paragraph, is equal to late start minus early start, or late finish minus early finish. In Figure 14, all the times for tasks are calculated by using forward pass then backward pass.



Note. The tasks having zero slack time create the critical path: A-C-D-E path.

Figure 14. Critical Path on Task Network Box Diagram

Another term that is important to understand is *slack*, or *float*, which must be known to easily find the critical path on a task network. Slack is the amount of extra time between the earliest and latest time of an event. Non-critical paths have slack times. For instance, if an activity takes three hours to complete, and there are six hours from start to the next activity, it means this activity has a three-hour slack. Besides, the slack time determines that the path on which a task exists is a non-critical path. In other words, critical events on a critical path have zero slack. Finding the slack makes it easy to find the critical path (Wiest & Levy, 1969). In our task network diagram, the A-B-D-E path is a non-critical path and it has a two-hour slack. In Figure 15, the slack time is illustrated on a Gantt chart for an example.

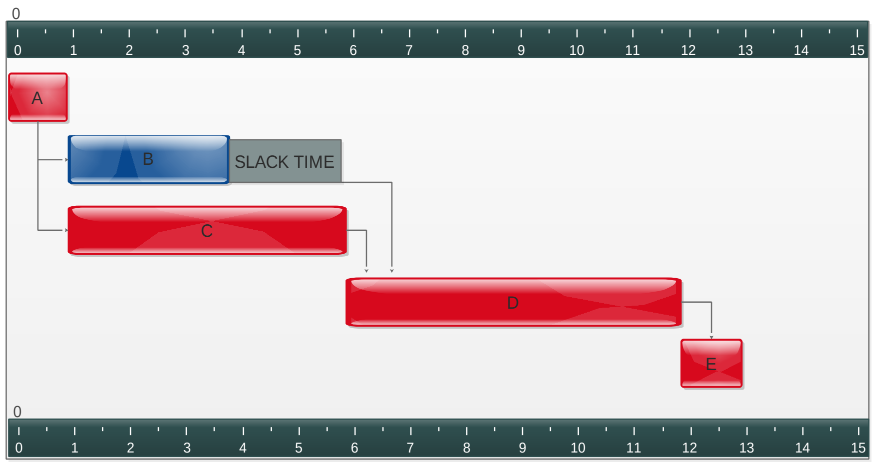


Figure 15. Slack Time on a Gantt Chart

There are two effective scheduling and management tools that can be utilized to find the critical path in military projects. These are the Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT). In the following paragraphs, these techniques are evaluated.

1. Critical Path Method

CPM is one of the best and useful techniques for planning, scheduling, and controlling a project. CPM assumes that the time required to complete individual tasks in a plan is known with certainty (Hattes, 1999). Because of that, this method may not be applicable to large and complex mission plans and projects. One of the main benefits of CPM is that it designates the shortest time needed to complete a plan. The main concept behind CPM is that you cannot start a new task before finishing some tasks. There are two types of tasks. The first are “sequential tasks,” which must be completed in a sequence; that is, each task must be finished before beginning the next tasks. The second type of tasks are “parallel tasks,” which are not based on the completion of any other tasks. They can be executed any time before or after a particular time is reached (Mind Tools Community, 2015).

2. Program Evaluation and Review Technique

PERT is a tool that serves the same functions as CPM; however, it differs from CPM in the completion times of every individual task. In PERT models, it is not possible to see a certainty in time like it is in CPM models. While creating the task list for this method, not only the most likely length of duration must be specified, as in CPM, but also the shortest possible duration and the longest possible duration of every task (Hillier & Lieberman, 1995). Even though it differs from CPM in terms of time certainty, it is usually used in conjunction with CPM. By considering the uncertainty in the nature of military operations, we can conclude that PERT is the best critical path analysis tool for this thesis. However, for this thesis, both tools are utilized to manage, schedule, and control the mission command plans. Table 3 shows three time estimates for each task.

Table 3. Task List for PERT Method

Task Name	Task Description	Optimistic Estimate	Typical Estimate	Pessimistic Estimate
A	Receive Warning Order	1 hour	2 hours	3 hours
B	Conduct Initial Mission Planning	3 hours	4 hours	5 hours
C	Occupy Positions	5 hours	6 hours	8 hours
D	Conduct Rehearsals	6 hours	8 hours	10 hours
E	Issue Execute Order	1 hour	2 hours	3 hours

The remainder of this thesis is organized as follows. In Chapter II, the scenario which is going to form the input data of the simulation will be designed as a task network diagram. In Chapter III, methodology and system design of the simulation will be explained in details. In Chapter IV, output data created after each replication of simulation will be analyzed. Lastly, in Chapter V, the main benefits of the simulation for decision-makers in their mission planning process will be discussed as a conclusion part.

THIS PAGE INTENTIONALLY LEFT BLANK

II. DATA AND SCENARIO

Using scenarios is a very efficient way to minimize the complexity of battle systems and help warfighters visualize situations in a battle. The scenario which is described in this chapter is designed to create suitable data for our simulation test-bed. It is purposefully organized to be short and simple so readers can understand what is going on in the simulation and see the whole picture easily. Because this scenario is not based on any known real or planned missions or capabilities, it should be considered unclassified. Lastly, this scenario is created according to the main principles of Mission Command Doctrine.

A. SITUATION

1st Brigade Combat Team has occupied an Area of Operation (AO) for conducting Wide Area Security Operations to defeat insurgents and to provide stability for the local population and government.

B. UNITS

1st Brigade Combat Team (1st BCT) is composed of the following forces:

1. Brigade Combat Team Headquarter (BCT HQ)
2. 1st Battalion Task Force (1st BNTF)
    - A Company (A CO) (-)
    - B Company (B CO)
    - C Company (C CO)
3. 2nd Battalion Task Force (2nd BNTF)
4. 3rd Battalion Task Force (3rd BNTF)
5. 1st Field Artillery 155 Self-Propelled Howitzer Battalion (1st FA BN)
    - A Field Artillery Battery (A FA BAT)
    - B Field Artillery Battery (B FA BAT)
    - C Field Artillery Battery (C FA BAT)
6. Sensors
    - Field Artillery Weapons Locating Radar (AN/TPQ-36)

- Fire Support Teams (FIST) at Company Level in Battalion Task Forces
- Unmanned Aircraft System (UAS)
- Unmanned Ground System (UGS)

C. ENEMY

Insurgents are operating across BCT AO with moderate population support targeting government operations such as police, military, and selective attacks on civilians supporting the Home Nation Government (HNG). According to initial intelligence reports, there is an insurgent group with 30 guerillas. They are armed with AK-47 Kalashnikovs, machine guns, RPG-7s, and 82-mm mortars.

D. WEATHER AND TERRAIN CONDITIONS

Weather and terrain conditions are benign and thus ignored in this scenario.

E. MISSION

Brigade HQ has issued an order to 1st BNTF. According to this order, one company of the 1st BNTF starts movement from base zone to AO at 1000 am and executes a patrolling mission there to find and neutralize insurgents and return to base before 0430 pm. A FA BAT supports this company in this mission with its fires due to intelligence that insurgents have 82-mm mortars and machine guns. With the order of the brigade commander, 1st FA BN assigns AN/TPQ-36 Radar and the 1st FIST to support this company. The brigade commander also assigns a UAV with its crew and a UGS with its crew to this mission.

F. TASKS

The brigade commander, after giving orders to the Battalion Task Force and Field Artillery Battalion commanders, assigns its G3 for planning and coordination of this mission. The commander wants his staff to conduct mission planning, simulate this plan, analyze it, and create a report before starting the mission. After getting this order, G3 decides to use the "Mission Planning

Simulation and Analyze Program.” In order to apply the appropriate planning data in the program, he asks FA BN S3, 1st BNTF S3, UGS team commander, and UAS team commander to estimate the time planned for subordinate units to accomplish tasks. He requests optimistic, pessimistic, and typical task durations for these tasks: planning, forward movement, patrolling, engagement, back movement, fire support, and intelligence support. The subordinate unit commanders give this information according to their experience and their mission rehearsal results. The details of the scenario are planned at company/battery level.

G. MISSION PLANNING

The subordinate unit commanders give the needed information to the brigade G3 according to their experience and their rehearsal results. The data taken from the subordinate commanders is in Tables 4–10.

Table 4. The Task Definitions and Task Durations for 1st BNTF

Number	Task Definition	Pessimistic Duration (min)	Typical Duration (min)	Optimistic Duration (min)
1	Receive Order from BDE	15	30	40
2	Plan Movement and Patrol of A CO	30	45	60

Table 5. The Task Definitions and Task Durations for 1st FA BN

Number	Task Definition	Pessimistic Duration (min)	Typical Duration (min)	Optimistic Duration (min)
1	Receive Order from BDE	15	30	40
2	Plan Movement and Patrol of A FA BAT	25	35	50

Table 6. The Task Definitions and Task Durations for A CO

Number	Task Definition	Pessimistic Duration (min)	Typical Duration (min)	Optimistic Duration (min)
1	Receive Order from 1st BNTF	8	15	25
2	Plan Movement and Patrol of Platoons	20	35	55
3	Forward Movement to AO	40	55	70
4	Patrol of AO	55	75	95
5	Engage with Insurgents and Neutralize Them	40	60	80
6	Back Movement to Base	45	65	75

Table 7. The Task Definitions and Task Durations for A CO FIST

Number	Task Definition	Pessimistic Duration (min)	Typical Duration (min)	Optimistic Duration (min)
1	Receive Order from A CO	8	15	25
2	Forward Movement to AO	40	55	70
3	Occupy Observing Position	15	25	35
4	Plan Fire Support for A CO	20	40	55
5	Detect Target and Call for Fire	10	18	30
6	Back Movement to Base	45	60	75

Table 8. The Task Definitions and Task Durations for A FA BAT

Number	Task Definition	Pessimistic Duration (min)	Typical Duration (min)	Optimistic Duration (min)
1	Receive Order from 1st FA BN	8	15	25
2	Plan Movement and Fire Support of Teams	30	45	60
3	Forward Movement to Fire Support Area	30	45	60
4	Prepare for Fire Support	25	35	45
5	Execute Fire Support	10	15	25
6	Back Movement to Base	35	50	65

Table 9. The Task Definitions and Task Durations for UAS

Number	Task Definition	Pessimistic Duration (min)	Typical Duration (min)	Optimistic Duration (min)
1	Receive Order from BDE	15	30	40
2	Prepare for Intelligence Support for A CO	30	40	55
3	Provide Intelligence	15	25	40

Table 10. The Task Definitions and Task Durations for UGS

Number	Task Definition	Pessimistic Duration (min)	Typical Duration (min)	Optimistic Duration (min)
1	Receive Order from BDE	15	30	40
2	Forward Movement to AO	35	50	65
3	Occupy Intelligence Support Position	20	30	40
4	Prepare for Intelligence Support for A CO	15	30	40
5	Provide Intelligence	10	20	30
6	Back Movement to Base	40	55	70

After getting all this information, G3 starts mission planning and putting all unit data and the data from BDE HQ into the empty task network list in an Excel sheet format. "Mission Planning Simulation and Analyze Program" will use this data from the task network Excel list. The G3 analyzes his mission plan by using the simulation program. The empty task network Excel list is shown in Table 11. The data elements consist of number, unit name, task name, task definition, optimistic duration, typical duration, pessimistic duration, and successor tasks.

Table 11. Empty Task Network List Table

Number	Unit Name	Task Name	Task Definition	Optimistic Duration (mins)	Typical Duration (mins)	Pessimistic Duration (mins)	Successor1	Successor2	Successor3	Successor4
0	Start	Start	Start	0	0	0	1			
	Finish	Finish	Finish	0	0	0				

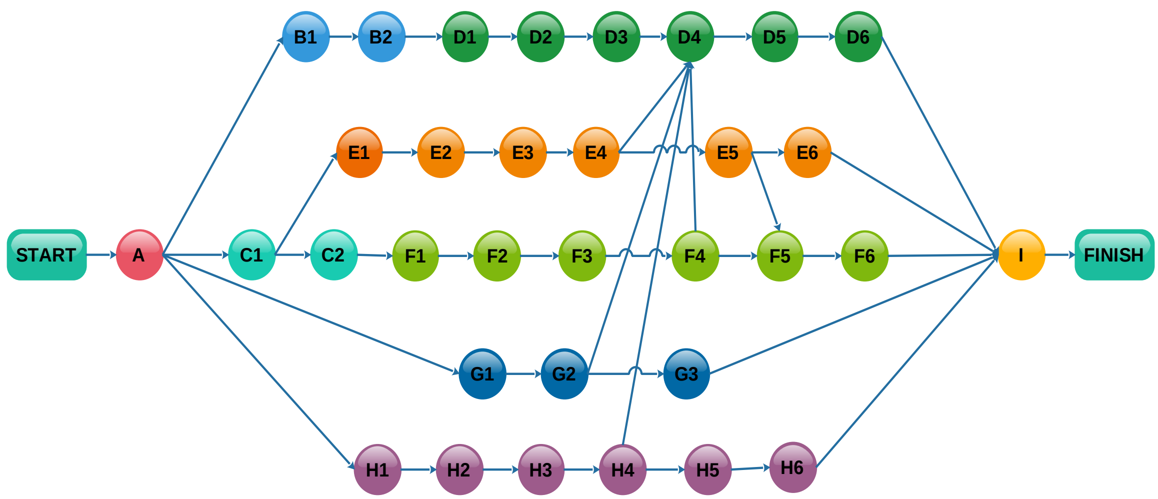
The G3 assigns a task name comprised of letters and numbers for all tasks. The most significant part of this planning is to select the successor tasks of needed tasks. By doing this, the G3 generates the task network of the mission plan. The filleds task network list is illustrated in Table 12.

Table 12. The Filled Task Network List of the Scenario

Number	Unit Name	Task Name	Task Definition	Optimistic Duration (mins)	Typical Duration (mins)	Pessimistic Duration (mins)	Successor1	Successor2	Successor3	Successor4
0	Start	Start	Start	0	0	0	1			
1	BDE HQ	A	Issue Order and Coordination of Mission	15	25	40	2	4	24	27
2	1st BNTF	B1	Receive Order from BDE	15	30	40	3			
3	1st BNTF	B2	Plan Movement and Patrol of A CO	30	45	60	6			
4	1st FA BN	C1	Receive Order from BDE	15	30	40	5	12		
5	1st FA BN	C2	Plan Movement and Patrol of A FA BAT	25	35	50	18			
6	A CO	D1	Receive Order from 1st BNTF	8	15	25	7			
7	A CO	D2	Plan Movement and Patrol of Platoons	20	35	55	8			
8	A CO	D3	Forward Movement to AO	40	55	70	9			
9	A CO	D4	Patrol of AO	55	75	95	10			
10	A CO	D5	Engage with Insurgents and Neutralize Them	40	60	80	11			
11	A CO	D6	Back Movement to Base	45	60	75	33			
12	A CO FIST	E1	Receive Order from A CO	8	15	25	13			
13	A CO FIST	E2	Forward Movement to AO	40	55	70	14			
14	A CO FIST	E3	Occupy Observing Position	15	25	35	15			
15	A CO FIST	E4	Plan Fire Support for A CO	20	40	55	9	16		
16	A CO FIST	E5	Detect Target and Call for Fire	10	18	30	17	22		
17	A CO FIST	E6	Back Movement to Base	45	60	75	33			
18	A FA BAT	F1	Receive Order from 1st FA BN	8	15	25	19			
19	A FA BAT	F2	Plan Movement and Fire Support of Teams	30	45	60	20			
20	A FA BAT	F3	Forward Movement to Fire Support Area	30	45	60	21			
21	A FA BAT	F4	Prepare for Fire Support	25	35	45	9	22		
22	A FA BAT	F5	Execute Fire Support	10	15	25	23			
23	A FA BAT	F6	Back Movement to Base	35	50	65	33			
24	UAS	G1	Receive Order from BDE	15	30	40	25			
25	UAS	G2	Prepare for Intelligence Support for A CO	30	40	55	9	26		
26	UAS	G3	Provide Intelligence	15	25	40	33			
27	UGS	H1	Receive Order from BDE	15	30	40	28			
28	UGS	H2	Forward Movement to AO	35	50	65	29			
29	UGS	H3	Occupy Intelligence Support Position	20	30	40	30			
30	UGS	H4	Prepare for Intelligence Support for A CO	15	30	40	9	31		
31	UGS	H5	Provide Intelligence	10	20	30	32			
32	UGS	H6	Back Movement to Base	40	55	70	33			
33	BDE HQ	I	Report to Brigade Commander	10	15	20	34			
34	Finish	Finish	Finish	0	0	0				

After that, the G3 can draw the task network diagram on a work sheet or use a program for drawing it. The best way to illustrate is to use the task names for all task nodes. The task network of the scenario consists of seven units and BCT headquarters. Each unit is represented with different colors. There are 33 different task nodes. The task network diagram is shown in Figure 16 for this scenario.

Figure 16. Task Network Diagram of Patrolling Mission Illustrated With Task Names



III. METHODOLOGY AND SYSTEM DESIGN

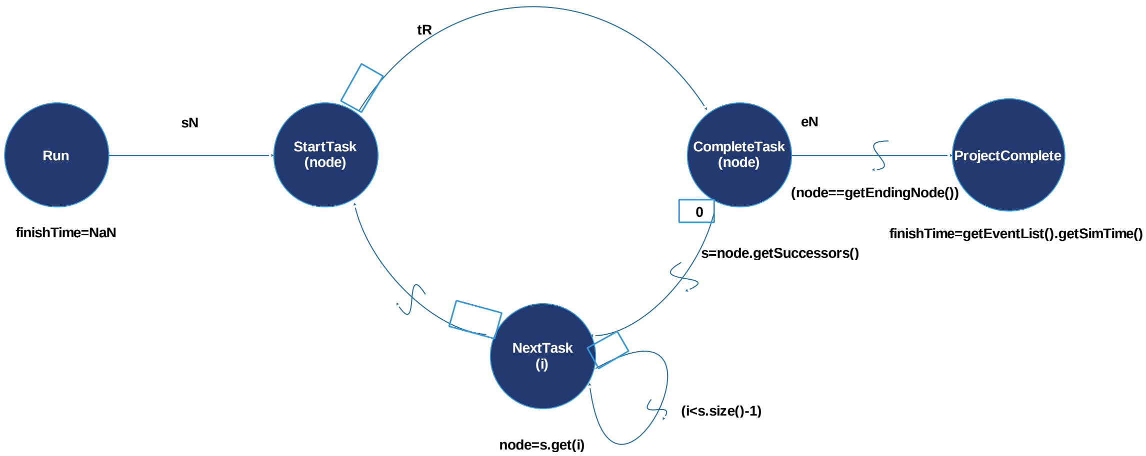
A. DISCRETE EVENT SIMULATION DESIGN

1. DES Components and Event Graphs

All of the components of the model that was created for running the simulation and analyzing the results were designed using Java software with Simkit, Apache POI, and Trac CPM libraries.

a. CPMComponent

CPMComponent class is the main component. It uses the Simkit library to create the main design of the simulation. The primary goal of this class is to determine the start time and finish time of all nodes and the project complete time of the project. To represent the nodes in our simulation “node” entities are created in doEvent() functions. This provides the ability to attach different time features to these entities. The doEvent() functions in this component are doStartTask(), doCompleteTask(), and doCompeteProject(). Of course, reset() and doRun() functions are also created in all other DES components. The event graph, parameters, and states of this component are shown in Figure 17.



Parameter	Type	Abbreviation
randomDuration	RandomVariate	tR
startingNode	Node	sN
endingNode	Node	eN

States	Explanation
finishTime	finishing time of all the simulation

Figure 17. Event Graph, Parameters, and States of CPMComponent

b. StartFinishListener

StartFinishListener class is also a simulation component which extends Simkit. As stated in the first chapter, the primary function of listener components is to make the events in one simulation component affect the state of another (Buss, 2011). Our listener component's main aim in our simulation is to affect some changes in the states of our main component, CPMComponent. The relationship of these two components is shown in Figure 18.

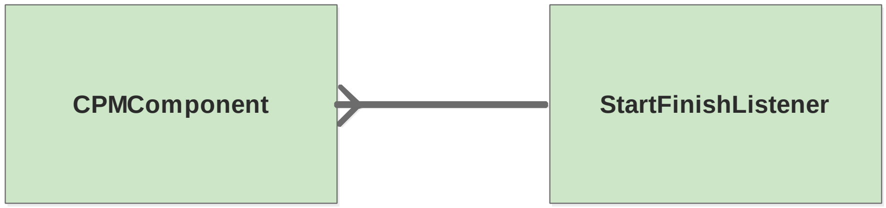


Figure 18. Relationship Between CPMComponent and StartFinishListener

The StartFinishListener component works as follows. The main events in our listener, doStartTask() and doCompleteTask(), have the same name as the events in the CPMComponent. This component listens to the events in the CPMComponent and logs the start and finish time of all nodes created in simulation. Figure 19 shows the event graph of StartFinishListener. There is no doRun() event in this listener. The StartFinishListener component's function is not to run the simulation, but to listen to the main component and capture data for analysis of our simulation.

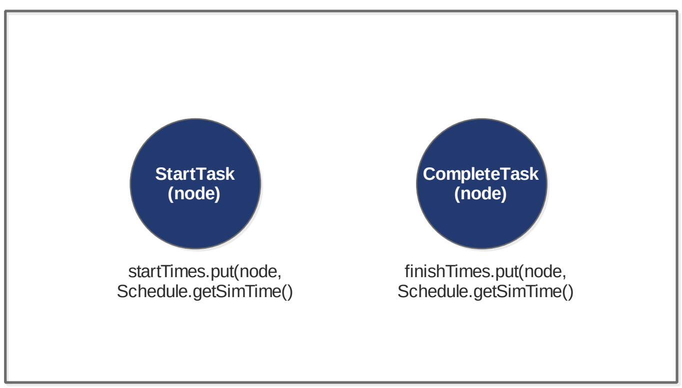


Figure 19. Event Graph of StartFinishListener

c. SimpleNode

SimpleNode is a component in which nodes are created with their names, task names, random duration, purpose, successors, predecessors, and states. One of the most important method of this class is toString() method. Using this method, we can see all the data of each task in the simulation.

d. Node

Node class is an interface class with all the methods in SimpleNode class. That is, SimpleNode implements the methods in the node class interface. The principal goal for using this class as an interface class is to encapsulate all the methods in one class.

e. NodeState

This component defines the states of the tasks. In this component, the enum function is used for assigning nodes one of three states: “not_started,” “underway,” and “completed.” These states are used in our main component’s methods to identify in which states the tasks are for a specific time interval.

2. Random Time Generator, Apache POI, and Runs

a. Random Variate Generator

There are many algorithms available to generate random variates from given probability distributions. These provide modelers with the basic tools which for implementing randomness in a simulation model (Buss, 2011). I use a triangle distribution as a random variate generation in Simkit for this simulation model. The triangle distribution is often used in business decision making and project management, especially in PERT models, to represent randomness in models in which the durations of the tasks are defined by a minimum and maximum value (“Triangular Distribution,” 2015). As mentioned in Chapter II, the decision-maker utilizes the optimistic, pessimistic, and typical duration estimates of the tasks for the military units. The “Mission Planning Simulation and Analyze Program” implements these three estimates in Simkit as a triangular distribution and creates random finishing times for every task node. As stated in CPMComponent class, while executing doStartTask() method, the finish time of this event is generated by RandomVariate according to the duration from a triangle distribution.

b. Apache POI

Apache POI is an open source Java library for Microsoft documents that allows programmers to create and display Office files using Java. Using methods and classes of this library, the user can write to Microsoft Office files and read from them easily. In our simulation, an Excel workbook is used to input our data and log the outputs of the simulation runs.

c. Run of the Simulation

To be able to get the required simulation outputs, we created three different scenarios, and for all these scenarios, three main classes are created to access the data input by the decision-maker in these scenarios. All the simulation code of the components and example code of one of the main classes are shown in Appendix A, B, C, D, E, and F.

B. DESCRIPTION OF THE SIMULATION AND ANALYZING THE MISSION PLANNING PROCESS

1. Mission Planning and Aims of the Simulation

In light of technological advancements in the military world, one of the aspects of war that should be paid the most attention is making fast and accurate mission plans. Despite the fact that plans may be revised several times during the course of war, the first plan must be prepared efficiently and with accuracy. Mission planning begins after receiving a mission from a higher command. Normally, the commander and his staff break down the mission into simple tasks to clarify the responsibilities of subordinates and to create a more understandable mission plan. In this thesis, the commander and his staff use Mission Command Concept to create their mission plan. The commander gives only the warning order and some relevant details about the operation to his subordinates, and wants them to take the initiative to plan and to provide three duration estimates for completing each task in their plan according to their experience and any rehearsal. The higher level commander and staff combine all the tasks taken from subordinates into a task network. The model and simulation in this thesis is a convenient program created in parallel to mission command approach. The simulation can help commanders and their staff in planning and analyzing missions by taking into consideration the contribution of technology and mission command to the speed and accuracy of the planned operations by focusing on the critical path of the task networks for the planned mission.

2. Outputs of the Simulation

By using the simulation described in this thesis, the commander and his staff will obtain the following three main outputs. They have the chance to reconsider their plan and revise it based on these outputs.

a. Success Percentage for Finishing the Mission on Time

After replicating the simulation several times, the mission planners can compare the commander's intent for the completion time of the mission with the completion time of the mission in simulation. The simulation gives a percentage for completing the mission on time. If the mission would not be completed on time, the mission planners have a chance to adjust the duration of the critical tasks by using some contribution of technology and mission command. They can then replicate the simulation to see whether the mission can be completed on time after adjustments.

b. The Critical Path of the Task Network

By finding the most likely critical path of the task network of the mission plan, mission planners have an idea of the shortest time to complete the mission and have an opportunity to focus on the tasks on the critical path, rather than all the tasks on the task network. If the critical path would not change after adjustments to the duration of the critical tasks on the critical path, the decision-makers have an opportunity to decide whether to use technology and mission command to shorten the times of other tasks to create a new critical path. They can evaluate how to use the personnel and other resources of the tasks with slack time for shortening the duration of the tasks on the critical path.

c. Start and Finish Time of Every Task Node

The start and finish time of task nodes are of importance while finding the critical path of the task network and also for allowing the synchronization of the simulation runs. As being one of outputs of the simulation, these times can be used for data analysis using statistical software such as R or JMP.

3. Task Network System

As mentioned in Chapter II, in this scenario, the mission is planned as a task network system. Every different task is represented with a node, and between two consecutive task nodes, there is an arc which represents the dependency for starting the task on one node and completion of the task on the prior node. The NPS thesis *Special Operations Mission Planning and Analysis Support System* by Keith A. Hattes (1999) formed a good starting point for this thesis. Although Hattes used a CPM approach rather than PERT to find the critical path in his code, we have used both approaches, and also made some alterations to the PERT approach in our simulation. As described in Chapter I, the main difference between CPM and PERT is whether the completion time of each task is a distribution rather than a fixed number. In CPM, the duration of all tasks is certain, while in PERT there are three duration estimates for calculating the expected time for all tasks. In our simulation, we use data from the triangle distribution to determine the finish time for each node rather than using the expected time equation of PERT. This allows us to account for variance.

4. Description of the Simulation

In Chapter II, our base scenario was created and data was entered into our empty Excel sheet, which is the input interface of the simulation. In the following paragraphs, we discuss which steps we took after running our simulation with the base scenario, how we created our new scenarios by taking the critical path created after the first run of the simulation into consideration, and what kind of process we followed to adjust the mission plan based on simulation results.

a. Replication of Scenario

The simulation is based on a stochastic simulation approach. We replicated the simulation 100 times for every scenario. The outputs are expected to be different for every replication because of the different start and finish times of each task node generated by the Random Variate Generator. As stated before, simulation creates three different kinds of output: Firstly, it gives the percentage of success of meeting the deadline of the mission determined by the commander before starting the mission; secondly, it provides the start and finish times of every task node; thirdly, it gives the most likely and alternative critical paths through the task network. After deciding a deadline for the mission and replication number of the simulation, the simulation gives the success percentage of finishing the mission on time. It is possible that the critical path may change for each replication of the simulation. Lastly, using the outputs of the start and finish times of all task nodes generated after every replication, we were able to analyze the outputs to see how task times affected the mission planning process.

b. Input and Output User Interface

The Excel file named "Mission Planning Simulation and Analyze Program" is both the input and output interface for our simulation. Users start the mission planning process with four Excel sheets called Deadline and Replication Sheet, InputSheetBase, InputSheetTech, and InputSheetMC, respectively. InputSheetBase, InputSheetTech, and InputSheetMC have an empty table for entering the scenario data. The empty table is shown in Table 13.

Table 13. Empty Task Network List Table

Number	Unit Name	Task Name	Task Definition	Optimistic Duration (mins)	Typical Duration (mins)	Pessimistic Duration (mins)	Successor1	Successor2	Successor3	Successor4
0	Start	Start	Start	0	0	0	1			
	Finish	Finish	Finish	0	0	0				

The Deadline and Replication Sheet also has a small table with two empty cells created for entering the replication number and deadline of the mission. The Deadline and Replication Sheet is shown in Table 14.

Table 14. Replication and Deadline Table

Deadline (minutes)	Replication Number of Simulation
390	100

After running the simulation for all scenarios, the program creates six new output Excel sheets called OutputSheetBase, OutputSheetTech, OutputSheetMC, BaseScenarioStat, TechScenarioStat, and MCScenarioStat respectively. The first three output sheets give the start and finish times of all task nodes with scenario name, run number, and node names. As mentioned before, this data is used to analyze the experiment in Chapter IV. The BaseScenarioStat, TechScenarioStat, and MCScenarioStat sheets give the success percentage of finishing the mission on time, the most likely and alternative critical paths from all scenario task networks.

c. Simulation Run with Base Scenario and Analysis of Outputs

Before the discussion of running the simulation with the base scenario and analyzing the outputs, we will briefly review the base scenario. As stated before, the commander intends to finish the mission in 6.5 hours (390 minutes) and to reach that goal 95% of the time. The G3 of the brigade has taken the three time estimates from the unit commanders, designed the base scenario as a task network in the empty task network table, and put this scenario in InputSheetBase. The base scenario is illustrated again in Table 15.

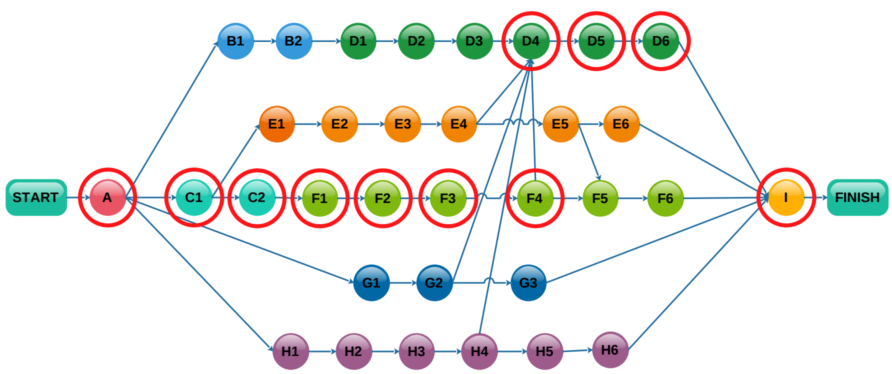
Table 15. The Task Network List of the Base Scenario

Number	Unit Name	Task Name	Task Definition	Optimistic Duration (mins)	Typical Duration (mins)	Pessimistic Duration (mins)	Successor1	Successor2	Successor3	Successor4
0	Start	Start	Start	0	0	0	1			
1	BDE HQ	A	Issue Order and Coordination of Mission	15	25	40	2	4	24	27
2	1st BNTF	B1	Receive Order from BDE	15	30	40	3			
3	1st BNTF	B2	Plan Movement and Patrol of A CO	30	45	60	6			
4	1st FA BN	C1	Receive Order from BDE	15	30	40	5	12		
5	1st FA BN	C2	Plan Movement and Patrol of A FA BAT	25	35	50	18			
6	A CO	D1	Receive Order from 1st BNTF	8	15	25	7			
7	A CO	D2	Plan Movement and Patrol of Platoons	20	35	55	8			
8	A CO	D3	Forward Movement to AO	40	55	70	9			
9	A CO	D4	Patrol of AO	55	75	95	10			
10	A CO	D5	Engage with Insurgents and Neutralize Them	40	60	80	11			
11	A CO	D6	Back Movement to Base	45	60	75	33			
12	A CO FIST	E1	Receive Order from A CO	8	15	25	13			
13	A CO FIST	E2	Forward Movement to AO	40	55	70	14			
14	A CO FIST	E3	Occupy Observing Position	15	25	35	15			
15	A CO FIST	E4	Plan Fire Support for A CO	20	40	55	9	16		
16	A CO FIST	E5	Detect Target and Call for Fire	10	18	30	17	22		
17	A CO FIST	E6	Back Movement to Base	45	60	75	33			
18	A FA BAT	F1	Receive Order from 1st FA BN	8	15	25	19			
19	A FA BAT	F2	Plan Movement and Fire Support of Teams	30	45	60	20			
20	A FA BAT	F3	Forward Movement to Fire Support Area	30	45	60	21			
21	A FA BAT	F4	Prepare for Fire Support	25	35	45	9	22		
22	A FA BAT	F5	Execute Fire Support	10	15	25	23			
23	A FA BAT	F6	Back Movement to Base	35	50	65	33			
24	UAS	G1	Receive Order from BDE	15	30	40	25			
25	UAS	G2	Prepare for Intelligence Support for A CO	30	40	55	9	26		
26	UAS	G3	Provide Intelligence	15	25	40	33			
27	UGS	H1	Receive Order from BDE	15	30	40	28			
28	UGS	H2	Forward Movement to AO	35	50	65	29			
29	UGS	H3	Occupy Intelligence Support Position	20	30	40	30			
30	UGS	H4	Prepare for Intelligence Support for A CO	15	30	40	9	31		
31	UGS	H5	Provide Intelligence	10	20	30	32			
32	UGS	H6	Back Movement to Base	40	55	70	33			
33	BDE HQ	I	Report to Brigade Commander	10	15	20	34			
34	Finish	Finish	Finish	0	0	0				

After reviewing the steps taken by decision-makers to form the base scenario, the next step of the simulation process is to enter the deadline as 390 minutes and replication number as 100. After entering all these inputs, the G3 runs the first scenario and controls the BaseScenarioStat sheet to see the mean critical path and deadline met percentage. Additionally, the G3 saves the data in the OutputSheetBase sheet for analyzing these data. Table 16 shows the data in the BaseScenarioStat sheet and the task network diagram with the most likely critical path is illustrated in Figure 20.

Table 16. BaseScenarioStat Output Table for Base Scenario With Deadline Met Percentage and Most Likely Critical Path

Deadline (min)	Deadline Met Percentage
390	0.0%
Most Likely Critical Path for Base Scenario	
Start	
A	
C1	
C2	
F1	
F2	
F3	
F4	
D4	
D5	
D6	
I	
Finish	



Note. The task nodes in red circles are critical task nodes.

Figure 20. Task Network Diagram Showing Most Likely Critical Path for Base Scenario

After seeing the deadline was met 0.0% of the time, the G3 focuses on the tasks on the critical path and uses critical thinking about how to decrease the task durations to meet the deadline 95% of the time. He analyzes critical tasks with the responsible unit leaders and reaches the solution that some critical task duration times can be accelerated by using high-tech equipment. In total, there are 11 critical tasks on the critical path; however, only five of their durations can be accelerated using high-tech equipment. Table 17 shows the tasks and measures taken to accelerate the durations.

Table 17. The Measures Taken to Accelerate the Durations of Critical Tasks

Unit Name	Task Name	Task Definition	The Measures Taken to Accelerate the Durations
A CO	D4	Patrol of AO	Using high-tech IFVs instead of old ones.
A CO	D5	Engage with Insurgents and Neutralize Them	Using high-tech IFVs instead of old ones.
A CO	D6	Back Movement to Base	Using high-tech IFVs instead of old ones.
A FA BAT	F3	Forward Movement to Fire Support Area	Using self-propelled howitzers instead of towed howitzers.
A FA BAT	F4	Prepare for Fire Support	Using self-propelled howitzers instead of towed howitzers. Using high-tech computer based systems for fire order calculation.

d. Simulation Run with Technology Scenario and Analysis of Outputs

After using some high-tech equipment in the operation rather than old equipment, the A CO commander and A FA BAT commander upgrade their three duration estimates according to the rehearsals they have done. The base scenario is revised according to these new durations and named Technology Scenario. This new scenario is shown in Table 18. The rows marked in yellow on Table 18 show the critical tasks whose duration estimates are changed.

Table 18. The Task Network List of the Technology Scenario

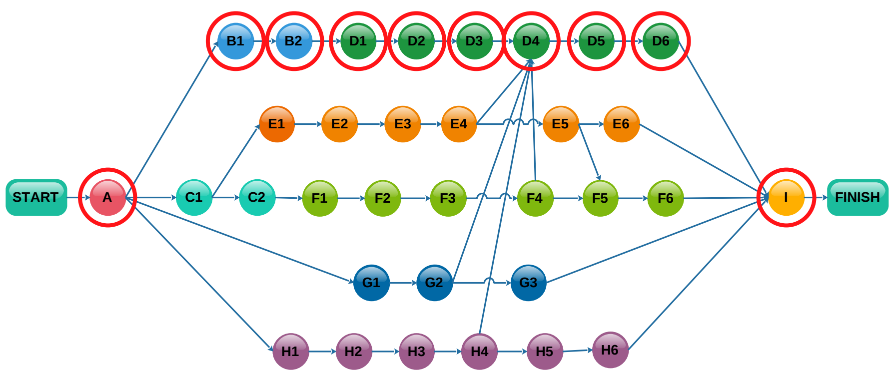
Number	Unit Name	Task Name	Task Definition	Optimistic Duration (mins)	Typical Duration (mins)	Pessimistic Duration (mins)	Successor1	Successor2	Successor3	Successor4
0	Start	Start	Start	0	0	0	1			
1	BDE HQ	A	Issue Order and Coordination of Mission	15	25	40	2	4	24	27
2	1st BNTF	B1	Receive Order from BDE	15	30	40	3			
3	1st BNTF	B2	Plan Movement and Patrol of A CO	30	45	60	6			
4	1st FA BN	C1	Receive Order from BDE	15	30	40	5	12		
5	1st FA BN	C2	Plan Movement and Patrol of A FA BAT	25	35	50	18			
6	A CO	D1	Receive Order from 1st BNTF	8	15	25	7			
7	A CO	D2	Plan Movement and Patrol of Platoons	20	35	55	8			
8	A CO	D3	Forward Movement to AO	40	55	70	9			
9	A CO	D4	Patrol of AO	40	60	75	10			
10	A CO	D5	Engage with Insurgents and Neutralize Them	25	45	65	11			
11	A CO	D6	Back Movement to Base	35	50	65	33			
12	A CO FIST	E1	Receive Order from A CO	8	15	25	13			
13	A CO FIST	E2	Forward Movement to AO	40	55	70	14			
14	A CO FIST	E3	Occupy Observing Position	15	25	35	15			
15	A CO FIST	E4	Plan Fire Support for A CO	20	40	55	9	16		
16	A CO FIST	E5	Detect Target and Call for Fire	10	18	30	17	22		
17	A CO FIST	E6	Back Movement to Base	45	60	75	33			
18	A FA BAT	F1	Receive Order from 1st FA BN	8	15	25	19			
19	A FA BAT	F2	Plan Movement and Fire Support of Teams	30	45	60	20			
20	A FA BAT	F3	Forward Movement to Fire Support Area	18	32	47	21			
21	A FA BAT	F4	Prepare for Fire Support	10	22	34	9	22		
22	A FA BAT	F5	Execute Fire Support	10	15	25	23			
23	A FA BAT	F6	Back Movement to Base	35	50	65	33			
24	UAS	G1	Receive Order from BDE	15	30	40	25			
25	UAS	G2	Prepare for Intelligence Support for A CO	30	40	55	9	26		
26	UAS	G3	Provide Intelligence	15	25	40	33			
27	UGS	H1	Receive Order from BDE	15	30	40	28			
28	UGS	H2	Forward Movement to AO	35	50	65	29			
29	UGS	H3	Occupy Intelligence Support Position	20	30	40	30			
30	UGS	H4	Prepare for Intelligence Support for A CO	15	30	40	9	31		
31	UGS	H5	Provide Intelligence	10	20	30	32			
32	UGS	H6	Back Movement to Base	40	55	70	33			
33	BDE HQ	I	Report to Brigade Commander	10	15	20	34			
34	Finish	Finish	Finish	0	0	0				

Note: The green rows show the tasks with new duration estimates.

After running the simulation again with the revised scenario with a deadline of 390 minutes and 100 replications, the simulation determines the most likely critical path and deadline met percentage shown in Table 19. Additionally, G3 saves the data in the OutputSheetTech sheet to use it in analysis. The task network diagram with the new critical path is illustrated in Figure 21.

Table 19. TechScenarioStat Output Table for Technology Scenario With Deadline Met Percentage and Most Likely Critical Path

Deadline (min)	**Deadline Met Percentage**
390	72.00%
Most Likely Critical Path for Tech Scenario	
Start	
A	
B1	
B2	
D1	
D2	
D3	
D4	
D5	
D6	
I	
Finish	



Note. The task nodes in red circles are critical task nodes.

Figure 21. Task Network Diagram Showing Critical Path for Technology Scenario

e. Simulation Run with Mission Command Scenario and Analysis of Outputs

According to the outputs of the simulation run with the technology scenario data, the percentage of replications in which the deadline was met increased from 0.0% to 72%. However, this is still not sufficient to meet the commander's intent of 95%. In addition, the critical path has changed, as shown in Figure 21.

The G3 evaluates the new situation and analyzes the critical tasks again with unit leaders. This time, there are 10 critical tasks on the critical path and five of them are the same tasks as on the base scenario's critical path. The durations of three of them have already upgraded with the technology input. After this analysis, the G3 suggests to the brigade commander that this time, the best option to accelerate the durations of critical tasks is to use the mission command concept. As described in Chapter I, the mission command concept gives subordinates the ultimate initiative and allows them to take risks during the course of war. Using initiative and taking calculated risks gives subordinates freedom of action and decreases the time used reporting all the details of every action by subordinates. The G3 takes new duration estimates revised with mission command from the unit commanders responsible for executing the critical tasks. The technology scenario is altered according to these new durations and named Mission Command Scenario. This new scenario is shown in Table 20. The rows marked with blue on Table 20 show the critical tasks whose duration estimates are changed.

Table 20. The Task Network List of the Mission Command Scenario

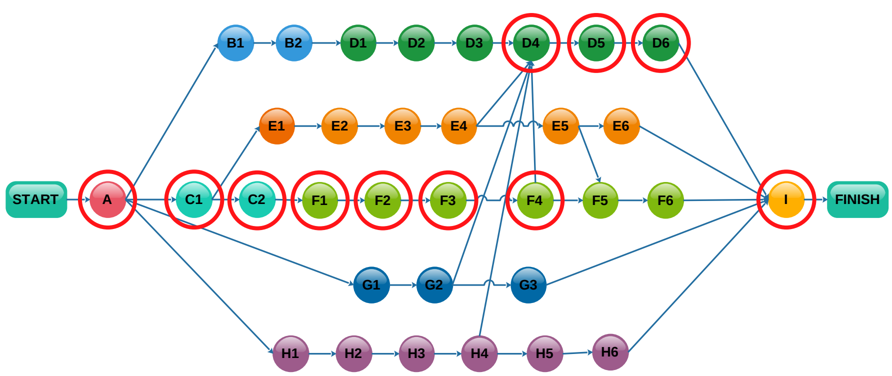
Number	Unit Name	Task Name	Task Definition	Optimistic Duration (mins)	Typical Duration (mins)	Pessimistic Duration (mins)	Successor1	Successor2	Successor3	Successor4
0	Start	Start	Start	0	0	0	1			
1	BDE HQ	A	Issue Order and Coordination of Mission	10	20	35	2	4	24	27
2	1st BNTF	B1	Receive Order from BDE	10	20	35	3			
3	1st BNTF	B2	Plan Movement and Patrol of A CO	25	40	55	6			
4	1st FA BN	C1	Receive Order from BDE	15	30	40	5	12		
5	1st FA BN	C2	Plan Movement and Patrol of A FA BAT	25	35	50	18			
6	A CO	D1	Receive Order from 1st BNTF	6	10	18	7			
7	A CO	D2	Plan Movement and Patrol of Platoons	15	30	45	8			
8	A CO	D3	Forward Movement to AO	30	40	55	9			
9	A CO	D4	Patrol of AO	30	50	65	10			
10	A CO	D5	Engage with Insurgents and Neutralize Them	20	40	55	11			
11	A CO	D6	Back Movement to Base	30	45	55	33			
12	A CO FIST	E1	Receive Order from A CO	8	15	25	13			
13	A CO FIST	E2	Forward Movement to AO	40	55	70	14			
14	A CO FIST	E3	Occupy Observing Position	15	25	35	15			
15	A CO FIST	E4	Plan Fire Support for A CO	20	40	55	9	16		
16	A CO FIST	E5	Detect Target and Call for Fire	10	18	30	17	22		
17	A CO FIST	E6	Back Movement to Base	45	60	75	33			
18	A FA BAT	F1	Receive Order from 1st FA BN	8	15	25	19			
19	A FA BAT	F2	Plan Movement and Fire Support of Teams	30	45	60	20			
20	A FA BAT	F3	Forward Movement to Fire Support Area	18	32	47	21			
21	A FA BAT	F4	Prepare for Fire Support	10	22	34	9	22		
22	A FA BAT	F5	Execute Fire Support	10	15	25	23			
23	A FA BAT	F6	Back Movement to Base	35	50	65	33			
24	UAS	G1	Receive Order from BDE	15	30	40	25			
25	UAS	G2	Prepare for Intelligence Support for A CO	30	40	55	9	26		
26	UAS	G3	Provide Intelligence	15	25	40	33			
27	UGS	H1	Receive Order from BDE	15	30	40	28			
28	UGS	H2	Forward Movement to AO	35	50	65	29			
29	UGS	H3	Occupy Intelligence Support Position	20	30	40	30			
30	UGS	H4	Prepare for Intelligence Support for A CO	15	30	40	9	31		
31	UGS	H5	Provide Intelligence	10	20	30	32			
32	UGS	H6	Back Movement to Base	40	55	70	33			
33	BDE HQ	I	Report to Brigade Commander	8	12	17	34			
34	Finish	Finish	Finish	0	0	0				

Note: The blue rows show the tasks with new duration estimates.

After running the simulation again with the revised scenario with a deadline of 390 minutes and 100 replications, the G3 reaches the most likely critical path and deadline met percentage shown in Table 21. Additionally, the G3 saves the data in the OutputSheetMC sheet to use them to analyze it in Chapter IV. The task network diagram with the new critical path is illustrated in Figure 22.

Table 21. MCScenarioStat Output Table for Technology Scenario With Deadline Met Percentage and Most Likely Critical Path

Deadline (min)	Deadline Met Percentage
390	99.00%
Critical Path for Mission Command Scenario	
Start	
A	
C1	
C2	
F1	
F2	
F3	
F4	
D4	
D5	
D6	
I	
Finish	



Note. The task nodes in red circles are critical task nodes.

Figure 22. Task Network Diagram Showing Critical Path for Mission Command Scenario

According to the outputs of the simulation run with the mission command scenario data, the deadline met percentage increases from 72.0% to 99.0%. At last, it meets the commander's intent of 95.0%. In addition, the most likely critical path has changed again. It is same as the most likely critical path of base scenario.

5. Conclusion of the Analysis of Runs

In this scenario, there is a total of 33 active task nodes excluding the start and finish nodes. The G3 has been able to focus on only 12 of them to meet the commander's intent. By doing this, he did not have to spend time on the other task nodes in order to speed up the mission duration, and did not have to use technology for the others either; thus, the cost of the mission did not increase, and resources were used more efficiently.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ANALYSIS OF DATA

As is mentioned in Chapter III, one output of the simulation is a list of start and finish times for all the task nodes for all 100 runs. This data is ready after running of each scenario. There are six output files in total, named OutputSheetBase, OutputSheetTech, OutputSheetMC, BaseScenarioStat, TechScenarioStat, and MCScenarioStat, respectively. The first three have 7,000 observations for statistical analysis, and the last three contain the critical paths of each replication. The R Statistical Program and Excel data analysis tools was used to analyze data after combining observations in a file named CombinedOutputData. Table 22 shows the first 20 rows of the data in the CombinedOutputData sheet as a table of data for an example.

Table 22. CombinedOutputData Sheet Example

Scenario	Run	Node	Event	Time
Base	1	Start	Start	0
Base	1	Start	Finish	0
Base	1	A	Start	0
Base	1	A	Finish	24.9247
Base	1	H1	Start	24.9247
Base	1	H1	Finish	55.55061
Base	1	C1	Start	24.9247
Base	1	C1	Finish	54.63778
Base	1	G1	Start	24.9247
Base	1	G1	Finish	57.13757
Base	1	B1	Start	24.9247
Base	1	B1	Finish	50.50483
Base	1	B2	Start	50.50483
Base	1	B2	Finish	89.34249
Base	1	C2	Start	54.63778
Base	1	C2	Finish	88.65859
Base	1	E1	Start	54.63778
Base	1	E1	Finish	70.36021
Base	1	H2	Start	55.55061

The data in Table 22 is in the form of a data set with each column a single variable, each row representing a single observation, and the total data set representing an observational unit. Scenario, run, node, and event are all factors, while time is the numeric value observed.

We will analyze the 21,000 rows of data using R and Microsoft Excel in order to:

- Analyze the different critical paths generated in replications of each scenario.
- Determine the effects of sample size on the mean and confidence interval of durations of each task node.

Before we analyze our data according to the goals mentioned above, we explore the data. We group data by scenario and run to focus on finish times for each task node for all 100 runs in each of three scenarios in Figure 23. Now we have an idea of how our random variate generator works using the triangle distribution for durations. As seen in Figure 23, nodes are arranged according to their finish time, and the variance of their finish time at the end of the task network is higher.

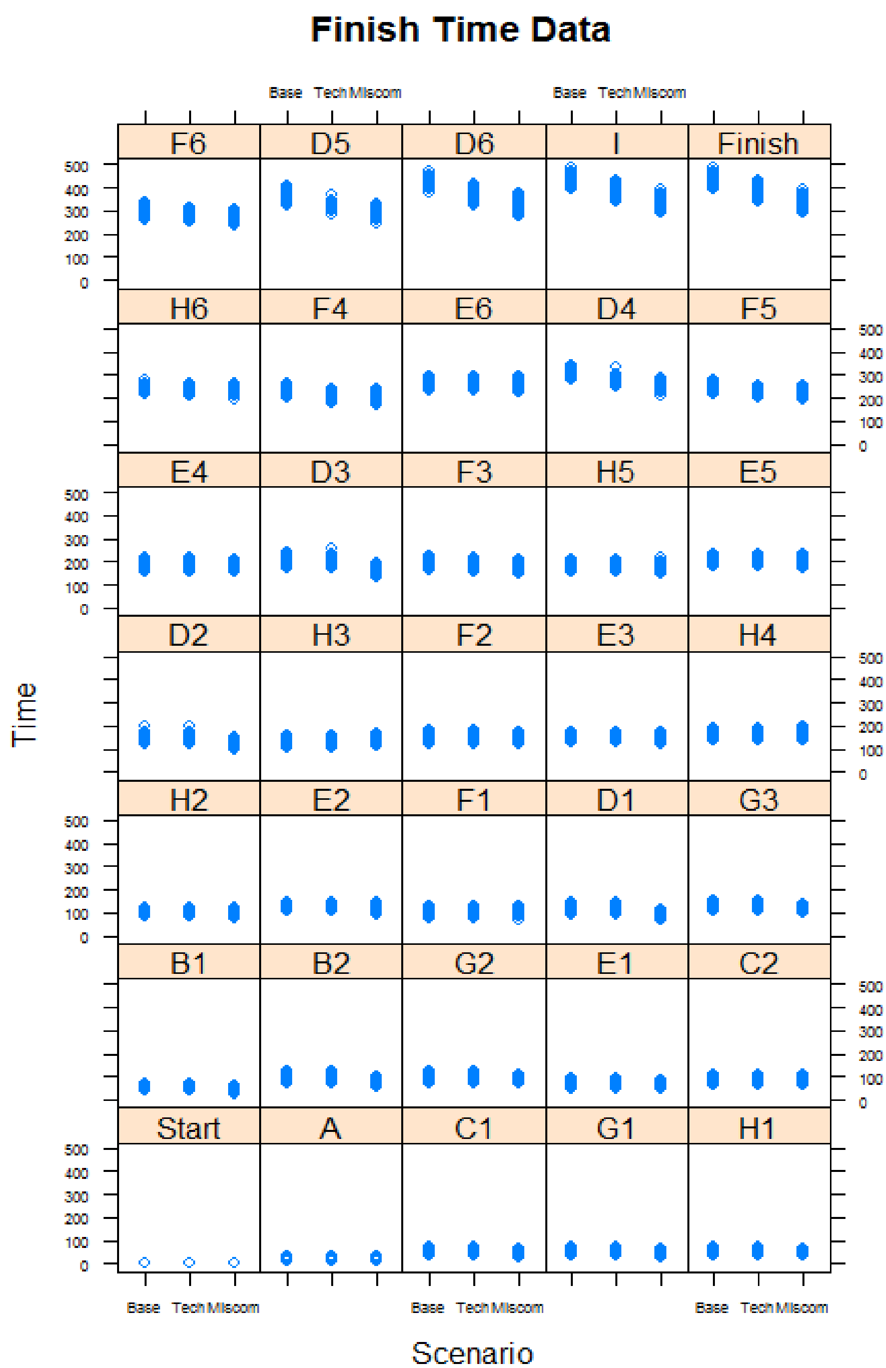


Figure 23. Finish Time Data of All Scenarios for 100 Runs of Simulation

A. ANALYZING CRITICAL PATHS OF SCENARIOS

In Chapter III, we discussed the most likely critical path of three scenarios created according to the mean durations of nodes. The decision-makers focused on these most likely critical paths for diminishing the durations of critical task nodes. However, due to the stochastic nature of the simulation, some replications of simulation generated critical paths different from the most likely critical path. In this section, we analyze all the critical paths of the three scenarios to determine whether the task nodes chosen to decrease the durations of tasks are appropriate or not.

1. Analyzing Critical Paths of Base Scenario

When we look at the BaseScenarioStat sheet, in addition to our most likely critical path and deadline met percentage results, we can see all critical paths generated for each replication. The critical path data in BaseScenarioStat sheet is shown in Appendix G. For this scenario, the most likely critical path was generated 95% of the time and it seems that it is appropriate to use it as our main critical path and decrease the durations of its task nodes to meet the commander's intent. The most likely critical path is shown in Figure 24. On the other hand, there is another critical path generated 5% of the time in the 100 replications. It is the most likely critical path of a technological scenario. The second most likely critical path is shown in Figure 25.

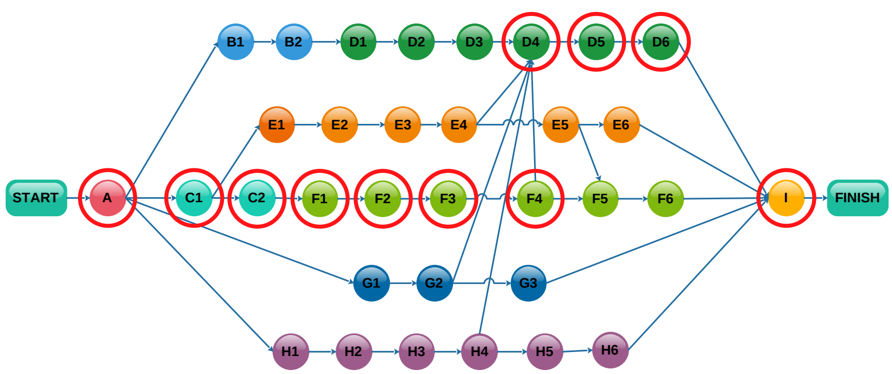


Figure 24. The Most Likely Critical Path Generated 95% of the Time in Base Scenario

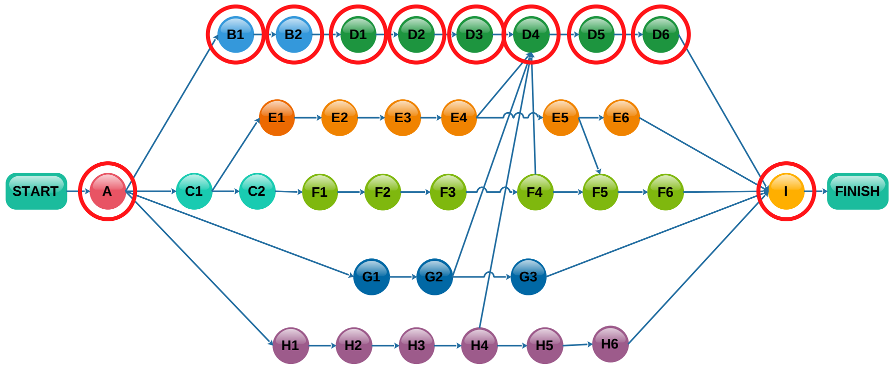


Figure 25. The Second Most Likely Critical Path Generated 5% of the Time in Base Scenario

When we compare the two critical paths in Figures 24 and 25, we see that the last four nodes (D4 through I) are the same. So, while decreasing the durations of critical nodes by using a technological contribution, it may be more efficient to focus on the same nodes in both critical paths and use technology as an accelarator. The second critical path has a big chance to be the most likely critical path of the next scenarios after adjustments in durations, thus it is likely cost-effective to use technology for these nodes.

2. Analyzing Critical Paths of Technology Scenario

When we look at the data about critical paths for this scenario found in TechScenarioStat sheet, we see that three different critical paths are generated. The critical path data in TechScenarioStat sheet is shown in Appendix H. For this scenario, the most likely critical path occurred 52% of the time, the second critical path occurred 44% of the time, and the last critical path occurred 4% of the time. The critical paths generated in this scenario are shown in the Figures 26, 27, and 28, respectively.

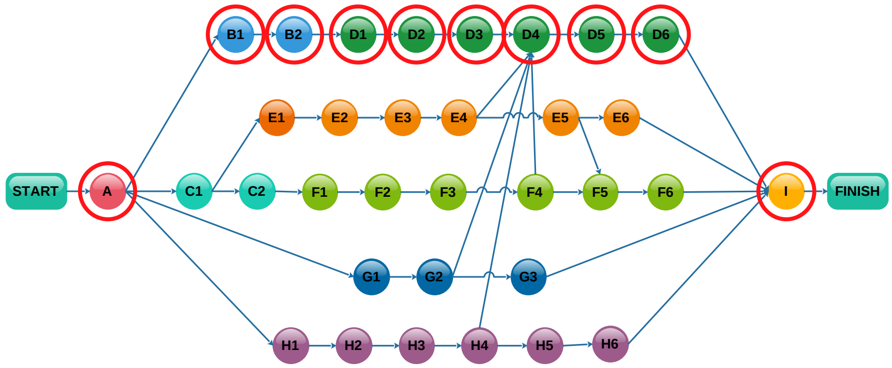


Figure 26. The Most Likely Critical Path Generated 52% of the Time in Technology Scenario

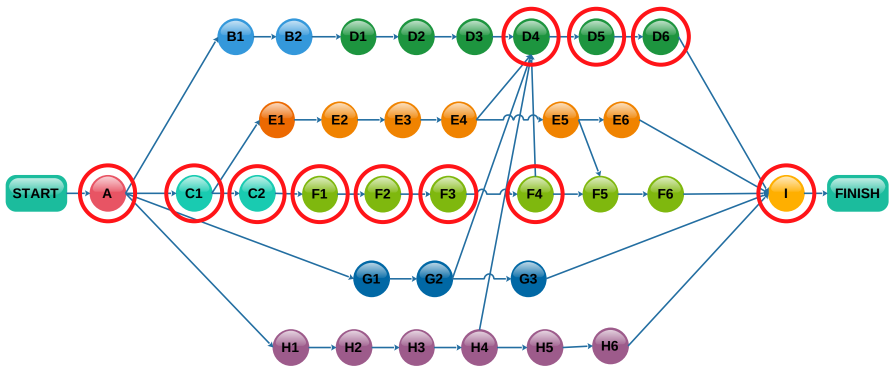


Figure 27. The Second Most Likely Critical Path Generated 44% of the Time in Technology Scenario

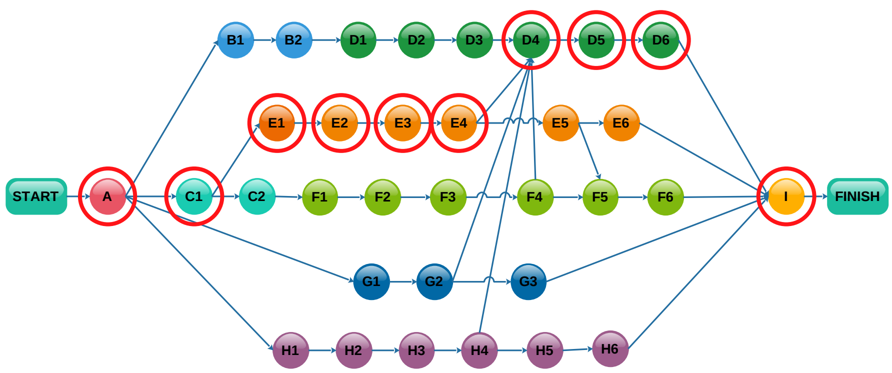


Figure 28. The Third Most Likely Critical Path Generated 4% of the Time in Technology Scenario

The last four nodes (D4 to I) of each critical path for this scenario are also the same as the base scenario. Thus, focusing on these nodes in all scenarios may be a very good option to decrease the durations of nodes to meet the commander's intent as well. However, in the first scenario, we improved the duration of these nodes by using technology. So, it was not likely to be a good idea to use technology again. Instead, we used the mission command concept to diminish the durations of these common nodes and other nodes on the most likely critical path. The second critical path is the same as the most likely critical path of the base scenario. Because the durations of the most likely critical path of base scenarios were adjusted before, it is not a good option to change the durations again, even if its generation percentage is as high as the most likely critical path of technology scenario. The third critical path of this scenario's generation percentage does not occur often, but still has a real chance to be the critical path of the last scenario. Because of that, focusing on the nodes shared by all critical paths and decreasing their durations may be a good idea.

3. Analyzing Critical Paths of Mission Command Scenario

When we look at the second output sheet of Mission Command Scenario, the MCScenarioStat sheet, we see that there are also three different critical paths. The critical path data in MCScenarioStat sheet is shown in Appendix I. However, the occurance percentage of one of them is much higher than the others and the focus is to decrease the overall duration of the mission. The most likely critical path occurs 87% of the time, the second most likely ciritical path occurs 12% of the time, and the third most likely critical path occurs 1% of the time for this scenario. The critical paths occurring in this scenario are shown in Figures 29, 30, and 31, respectively.

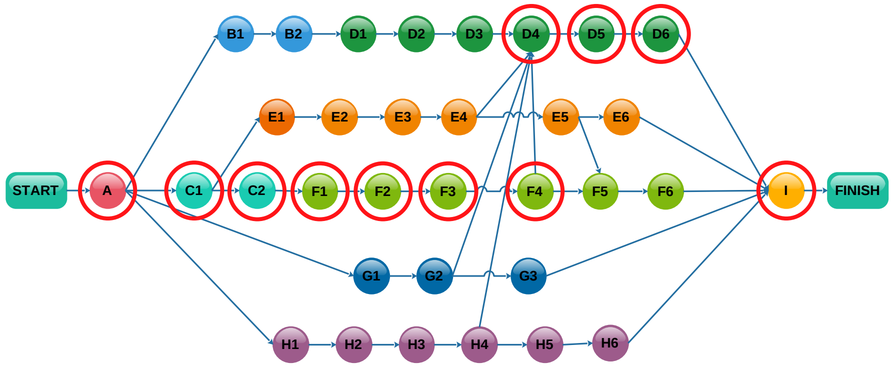


Figure 29. The Most Likely Critical Path Generated 87% of the Time in Mission Command Scenario

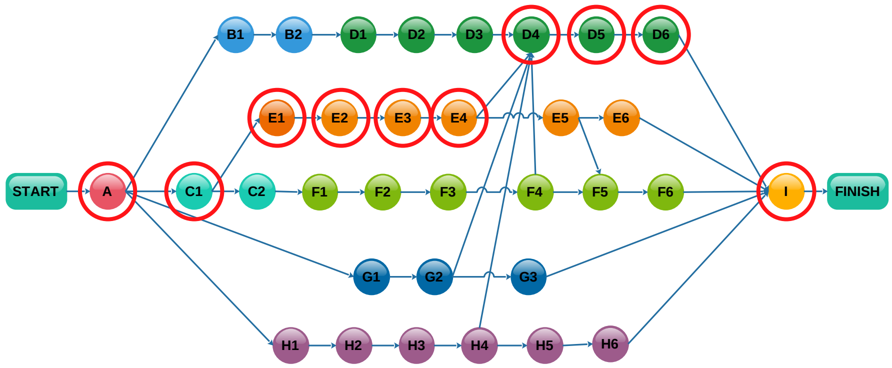


Figure 30. The Second Most Likely Critical Path Generated 12% of the Time in Mission Command Scenario

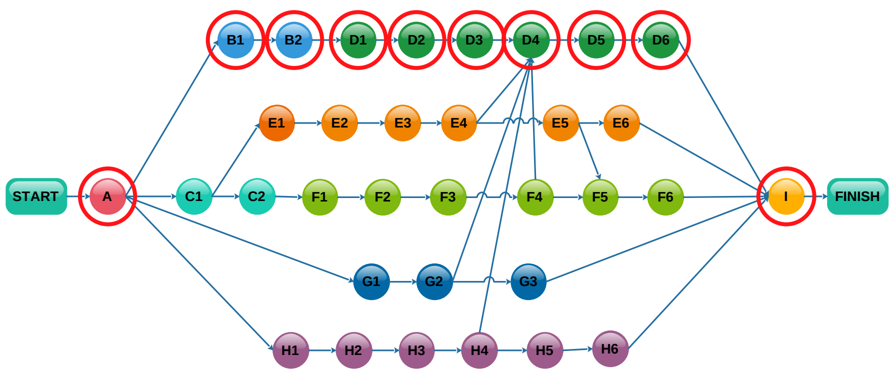


Figure 31. The Third Most Likely Critical Path Generated 1% of the Time in Mission Command Scenario

The three critical paths of this scenario are the same as the critical paths of the technology scenario with different occurance percentages. Since the last scenario of the simulation meets the commander's intent for overall mission duration, there is no need to attempt to reduce the durations of the most likely critical path nodes. The second most likely critical path has a significant chance to be the first most likely critical path for next replications and its nodes were not adjusted before. So, the contribution of technology and mission command might be used on these nodes.

Figure 32 depicts the distribution of the mean finish time for each node. When we look at Figures 29, 30, and 31, we see that the nodes D3, F4, and E4 are the predecessor nodes of common nodes of the critical paths for all three scenarios. So, the distribution of the mean finish time of these nodes gives us insight into the occurrence of critical paths of all the scenarios.

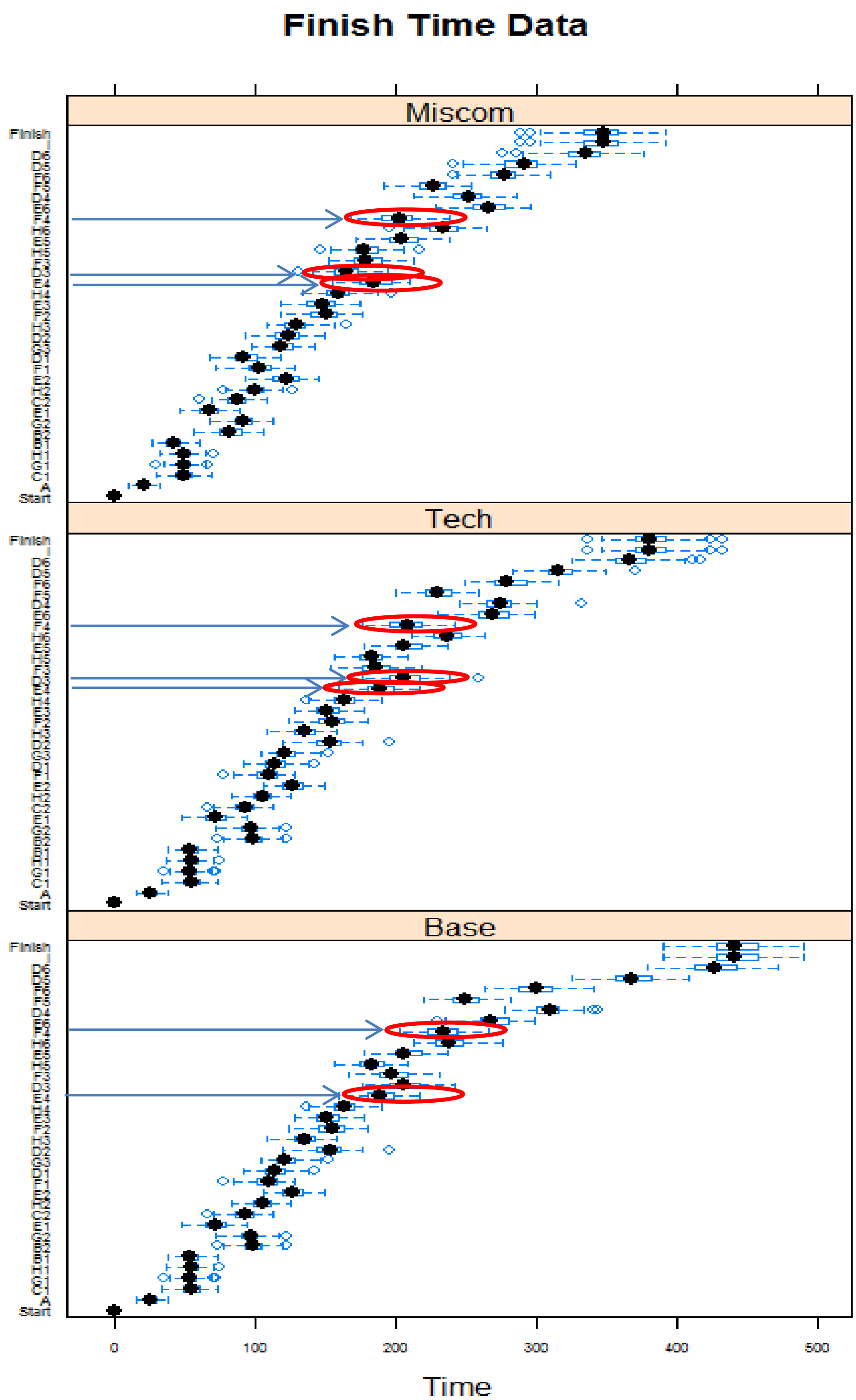


Figure 32. Mean and Distribution of Finish Times of Task Nodes

As seen in Figure 32, for the base scenario, the mean finish time of F4 on the most likely critical path is higher than D3 on the second critical path. This explains the high percentage of occurrence of this node on the most likely critical path. When we look at the data for technology scenario, it is also seen that the mean and confidence interval of the finish time of nodes D3 and F4 are almost the same. It shows the equal percentage of occurrence of the critical paths through these nodes. Lastly, the situation in the mission command scenario is the same as that of other scenarios. The mean finish time of the F4 node is higher than others, so the occurrence of this node on the critical path is expected as well.

B. COMPARING MEANS AND CONFIDENCE INTERVALS OF TASK DURATIONS TO SEE THE IMPACT OF SAMPLE SIZE

As stated before, the three durations used in input sheets are estimated by the unit commanders according to their experience and rehearsals made for specific tasks. After putting these durations into a triangle distribution, it is expected that the random variate generator generates random node durations with a mean close to the mean of these durations with sufficient replications. In this section, we compare the mean of the durations estimated by unit commanders and the mean of random durations generated by the triangle distribution to see the impacts of increasing the sample size of replications on the random durations and to verify that 100 replications is sufficient.

Table 23 includes the means of durations used in triangle distribution, mean durations generated after 100 replications and 500 replications of simulation, and their 95% confidence intervals for the task nodes of Base Scenario.

Table 23. Comparison Table of Means and Confidence Intervals of Durations of Nodes

Number	Task Name	Mean of Durations for Triangle Distribution	Mean Duration of 100 Replication	Mean Duration of 500 Replication	CI95 Lower 100 Replication	CI95 Upper 100 Replication	CI95 Lower 500 Replication	CI95 Upper 500 Replication
1	A	26.67	26.06	26.78	25.77	27.79	25.10	27.03
2	B1	28.33	27.80	28.28	27.25	29.31	26.74	28.87
3	B2	45.00	45.03	45.80	44.63	46.97	43.84	46.23
4	C1	28.33	28.94	27.95	26.94	28.96	27.94	29.94
5	C2	36.67	37.19	36.75	35.71	37.78	36.12	38.26
6	D1	16.00	15.70	16.17	15.46	16.88	15.03	16.37
7	D2	36.67	37.38	36.80	35.40	38.19	35.97	38.78
8	D3	55.00	54.14	55.04	53.85	56.22	52.94	55.34
9	D4	75.00	74.90	75.51	73.94	77.07	73.33	76.46
10	D5	60.00	59.32	59.79	58.19	61.40	57.61	61.03
11	D6	60.00	59.65	60.04	58.78	61.31	58.41	60.90
12	E1	16.00	16.84	16.12	15.42	16.82	16.14	17.55
13	E2	55.00	54.65	54.75	53.54	55.97	53.32	55.99
14	E3	25.00	24.17	25.22	24.37	26.07	23.31	25.04
15	E4	38.33	37.97	38.46	37.07	39.85	36.52	39.42
16	E5	19.33	18.62	19.31	18.52	20.10	17.86	19.38
17	E6	60.00	61.07	60.13	58.94	61.31	59.83	62.32
18	F1	16.00	16.09	16.01	15.31	16.71	15.39	16.78
19	F2	45.00	45.39	45.35	44.13	46.57	44.25	46.52
20	F3	45.00	44.05	44.66	43.40	45.91	42.88	45.22
21	F4	35.00	34.85	34.89	34.07	35.71	34.10	35.60
22	F5	16.67	16.50	16.76	16.14	17.38	15.90	17.10
23	F6	50.00	49.86	49.80	48.61	50.99	48.80	50.92
24	G1	28.33	28.09	28.38	27.31	29.46	27.09	29.09
25	G2	41.67	42.00	41.96	40.94	42.99	40.97	43.04
26	G3	26.67	25.68	26.62	25.61	27.63	24.78	26.58
27	H1	28.33	28.54	28.12	27.12	29.12	27.48	29.60
28	H2	50.00	49.55	49.81	48.56	51.06	48.24	50.87
29	H3	30.00	30.35	30.24	29.44	31.05	29.47	31.24
30	H4	28.33	28.63	28.25	27.19	29.30	27.56	29.71
31	H5	20.00	20.01	19.90	19.08	20.72	19.27	20.75
32	H6	55.00	54.39	55.09	53.95	56.24	53.24	55.53
33	I	15.00	14.90	14.91	14.50	15.32	14.48	15.33

When we compare the means of unit commanders' estimated durations (they are also the mean durations of the triangle distribution) and the mean durations of 100 replications and 500 replications, we can see that there is not a significant difference between them. This shows that the unit commanders' duration estimates are reproduced by the simulation, and increasing the sample size beyond 100 does not make a difference. With 100 runs, the confidence interval converges sufficiently. Making 500 runs does not reduce the confidence interval significantly and sometimes the confidence interval increases slightly. Thus the difference in confidence intervals for a given node between 100 runs and 500 runs is noise, and 100 runs are sufficient to analyze the task network.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

Time is of importance for the success of a military mission and is a beneficial tool that will provide an advantage over the enemy if used effectively. This thesis provides a mission planning and analysis simulation for commanders and staff officers in their mission planning process and helps them to manage the duration of a mission by using technology and mission command as a means to accelerate selected tasks. The simulation and analysis of the output data in this thesis has five main benefits in helping military decision-makers during mission planning. Firstly, it finds the critical path of the task network system of a mission planned according to the mission command concept. Secondly, it gives the estimated success rate of finishing the mission on time. Thirdly, it provides an interactive mission planning process and amortizes most critical factors of war such as morale, leadership, weather, and terrain. Fourthly, it saves the time and money spent for the mission by focusing on only a small part of the tasks planned during the mission plan process. Lastly, the excel sheets can be used as a matrix for synchronizing the mission plan.

As was discussed previously, finding the critical path in a task network is very useful in estimating the finishing time of the mission, because the critical path is the longest path in a task network and determines the duration of the mission. Although the scenario considered here was small, in general task networks can be quite large. In a larger task network system, the decision-makers cannot focus on all the tasks. Thus, finding and focusing on the critical path finding makes sense. In our scenario, the G3 of the brigade was able to concentrate on only 30% of the tasks for shortening the duration of the mission using technology and mission command to accelerate tasks. In the end, the G3 met the commander’s intent of achieving a very high probability of finishing the mission on time.

The simulation gives decision-makers a good idea of the likelihood of finishing the mission on time. In our scenario, the commander’s intent is to finish the mission on time 95% of the time. Thanks to the stochastic approach in the simulation, the G3 could adjust the deadline and replication number of the simulation. After replicating the simulation according to the replication number, the G3 saw the success rate percentage and determined whether the mission would likely finish on time or not.

One of the most important advantages of this simulation is interaction in the mission planning process. It is not only the decision-makers such as the G2 and BDE commander, but also the unit commanders who participate in the mission planning process. This approach forces lower units on planning process by taking their time estimates of the tasks they will perform. They become familiar with all the details of the mission and gain the ability to use initiative during unexpected situations that occur in operations. Thus, they are trained during the planning phase using mission command principles. Lastly, although some significant factors such as morale, leadership, weather, and terrain are not represented directly in the simulation, their impacts are estimated by unit commanders and shown indirectly in their duration estimates.

The simulation shows that it would not shorten the duration of a mission if technological enhancements were supplied to the tasks apart from the tasks on the critical path. This saves time and resources when planning the mission. Nevertheless, decision-makers should consider that using the technology contribution to reduce the mission duration is not applicable to all tasks on the critical path due to the structure of some tasks. For instance, while giving a warning order, technology is not needed. In addition, the slack time and manpower saved from the non-critical tasks can be utilized for the success of other tasks in the same mission and even for following missions. When we look at the time savings generated by applying the mission command concept to the mission, the analysis is similar. There is no need to apply mission command to reduce the duration of non-critical tasks. However, it is required to evaluate the status of all the critical tasks comprehensively before using mission command to save time. Even if mission command is a very efficacious concept for shortening mission durations and for the overall success of missions, decision-makers must consider some tasks in which it cannot be used. For instance, some critical tasks in counter-insurgency operations require detailed plans.

Lastly, the Excel sheets can be used as synchronization matrices for mission planners while planning the mission, coordinating the operation, and following the movements of units. There is no need for another platform to develop and visualize the time schedules and planning matrices.

In conclusion, this thesis provides a simulation that can be utilized as a mission planning and analysis tool. The system architecture of the simulation can be modified and improved for meeting other needs in the mission command concept and mission planning process in the future.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. CPMComponent.java

```
package edu.nps.moves.network;

import static edu.nps.moves.network.NodeState.COMPLETED;
import static edu.nps.moves.network.NodeState.NOT_STARTED;
import static edu.nps.moves.network.NodeState.UNDER_WAY;
import static java.lang.Double.NaN;
import simkit.SimEntityBase;

/**
 *
 * @author hasanbeker
 */
public class CPMComponent extends SimEntityBase {

    private Node startingNode; // parameter

    private Node endingNode; // parameter

    protected double finishTime; // state

    public CPMComponent() {
    }

    public CPMComponent(Node startingNode, Node endingNode) {
        this.setStartingNode(startingNode);
        this.setEndingNode(endingNode);
    }

    @Override
    public void reset() {
        super.reset();
        this.finishTime = NaN;
    }

    public void doRun() {
        waitDelay("StartTask", 0.0, getStartingNode());
    }

    public void doStartTask(Node node) {
        node.setState(UNDER_WAY);
        firePropertyChange("node", node);

        node.setDuration(node.getRandomDuration().generate());
        waitDelay("CompleteTask", node.getDuration(), node);
    }

    public void doCompleteTask(Node node) {
        node.setState(COMPLETED);
        firePropertyChange("node", node);

        for (Node successor : node.getSuccessors()) {
            if (successor.canStart()) {
                waitDelay("StartTask", 0.0, successor);
            }
        }
```

```
        if (node == getEndingNode()) {
            waitDelay("ProjectComplete", 0.0);
        }
    }

    public void doProjectComplete() {
        finishTime = getEventList().getSimTime();
        firePropertyChange("finishTime", getFinishTime());

        waitDelay("StartTaskReverse", 0.0, getEndingNode());
    }

    public void doStartTaskReverse(Node node) {
        node.setState(UNDER_WAY);
        firePropertyChange("node", node);

        waitDelay("CompleteTaskReverse", node.getDuration(), node);
    }

    public void doCompleteTaskReverse(Node node) {
        node.setState(NOT_STARTED);
        firePropertyChange("node", node);

        for (Node predecessor : node.getPredecessors()) {
            if (canStartReverse(node)) {
                waitDelay("StartTaskReverse", 0.0, predecessor);
            }
        }
    }

    public static boolean canStartReverse(Node node) {
        for (Node successor: node.getSuccessors()) {
            if (node.getState() != NOT_STARTED) {
                return false;
            }
        }
        return true;
    }

    public Node getStartingNode() {
        return startingNode;
    }

    public void setStartingNode(Node startingNode) {
        this.startingNode = startingNode;
    }

    public Node getEndingNode() {
        return endingNode;
    }

    public void setEndingNode(Node endingNode) {
        this.endingNode = endingNode;
    }

    public double getFinishTime() {
        return finishTime;
    }

}
```

APPENDIX B. Node.java

```
package edu.nps.moves.network;

import java.util.List;
import simkit.random.RandomVariate;

/**
 *
 * @author hasanbeker
 */
public interface Node {

    public String getName();

    public NodeState getState();

    public void setState(NodeState state);

    public String getTaskName();

    public String getPurpose();

    public void setDuration(double duration);

    public double getDuration();

    public List<Node> getPredecessors();

    public List<Node> getSuccessors();

    public void addPredecessor(Node predecessor);

    public void addSuccessor(Node successor);

    public boolean canStart();

    public RandomVariate getRandomDuration();

}
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. NodeState.java

```
package edu.nps.moves.network;

/**
 *
 * @author hasan beker
 */
public enum NodeState {
   NOT_STARTED,
   UNDER_WAY,
   COMPLETED
}
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. SimpleNode.java

```
package edu.nps.moves.network;

import static edu.nps.moves.network.NodeState.COMPLETED;
import static edu.nps.moves.network.NodeState.NOT_STARTED;
import java.util.ArrayList;
import java.util.List;
import java.util.SortedSet;
import java.util.TreeSet;
import simkit.random.RandomVariate;

/**
 *
 * @author hasanbeker
 */
public class SimpleNode implements Node, Comparable<Node> {

    private final String name;

    private NodeState state;

    private String taskName;

    private final String purpose;

    private double duration;

    private RandomVariate randomDuration;

    private final SortedSet<Node> predecessors;

    private final SortedSet<Node> successors;

    public SimpleNode(String name, String taskName, String purpose, RandomVariate randomDuration) {

        this.setRandomDuration(randomDuration);
        this.name = name;
        this.taskName = taskName;
        this.purpose = purpose;
        this.setDuration(duration);
        this.state = NOT_STARTED;
        this.predecessors = new TreeSet<>();
        this.successors = new TreeSet<>();

    }

    @Override
    public String getName() {
        return name;
    }

    @Override
    public NodeState getState() {
        return state;
    }

    @Override
```

```
public void setState(NodeState state) {
    this.state = state;
}

@Override
public String getTaskName() {
    return taskName;
}

@Override
public String getPurpose() {
    return purpose;
}

@Override
public double getDuration() {
    return duration;
}

@Override
public List<Node> getPredecessors() {
    return new ArrayList<>(predecessors);
}

@Override
public List<Node> getSuccessors() {
    return new ArrayList<>(successors);
}

@Override
public int compareTo(Node o) {
    return this.hashCode() - o.hashCode();
}

public void setDuration(double duration) {
    if (duration < 0.0) {
        throw new IllegalArgumentException(
                String.format("duration must be \uu2265 0.0: %.1f", duration));
    }
    this.duration = duration;
}

@Override
public void addPredecessor(Node predecessor) {
    if (this.predecessors.add(predecessor)) {
        predecessor.addSuccessor(this);
    }
}

@Override
public void addSuccessor(Node successor) {
    if (this.successors.add(successor)) {
        successor.addPredecessor(this);
    }
}

@Override
public String toString() {
    return String.format("Unit Name: %s - Task Name: '%s' - Purpose: %s - State: %s - Duration: %.1f min",
                                    getName(), getTaskName(), getPurpose(),getState(), getDuration());
}
```

```
    @Override
    public boolean canStart() {
        for (Node predecessor : predecessors) {
            if (predecessor.getState() != COMPLETED) {
                return false;
            }
        }
        return true;
    }

    @Override
    public RandomVariate getRandomDuration() {
        return randomDuration;
    }

    public void setRandomDuration(RandomVariate randomDuration) {
        this.randomDuration = randomDuration;
    }

}
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. StartFinishListener.java

```
package edu.nps.moves.network;

import static java.lang.Math.abs;
import java.util.LinkedHashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import simkit.Schedule;
import simkit.SimEntityBase;

/**
 *
 * @author hasanbeker
 */
public class StartFinishListener extends SimEntityBase {

    public static final double EPSILON = 1.0E-10;

    protected Map<Node, Double> earlyStartTimes;

    protected Map<Node, Double> earlyFinishTimes;

    protected Map<Node, Double> lateStartTimes;

    protected Map<Node, Double> lateFinishTimes;

    protected Map<Node, Double> slack;

    protected double finishTime;

    public StartFinishListener() {
        earlyStartTimes = new LinkedHashMap<>();
        earlyFinishTimes = new LinkedHashMap<>();
        lateStartTimes = new LinkedHashMap<>();
        lateFinishTimes = new LinkedHashMap<>();
        slack = new LinkedHashMap<>();
    }

    @Override
    public void reset() {
        super.reset();
        earlyStartTimes.clear();
        earlyFinishTimes.clear();
        lateStartTimes.clear();
        lateFinishTimes.clear();
        slack.clear();
        finishTime = 0.0;
    }

    public void doStartTask(Node node) {
        earlyStartTimes.put(node, Schedule.getSimTime());
    }

    public void doCompleteTask(Node node) {
        earlyFinishTimes.put(node, Schedule.getSimTime());
    }
```

```java
    public void doStartTaskReverse(Node node) {
        lateFinishTimes.put(node, 2.0 * getFinishTime() - Schedule.getSimTime());
    }

    public void doCompleteTaskReverse(Node node) {
        lateStartTimes.put(node, 2.0 * getFinishTime() - Schedule.getSimTime());
        slack.put(node, abs(earlyStartTimes.get(node) - lateStartTimes.get(node)));
    }

    public void doProjectComplete() {
        finishTime = Schedule.getSimTime();
    }

    public List<Node> getCriticalPath(Node startingNode, Node endingNode) {
        List<Node> criticalPath = new LinkedList<>();
        Node node = startingNode;
        do {
            criticalPath.add(node);
            for (Node successor : node.getSuccessors()) {
                if (slack.get(successor) < EPSILON) {
                    node = successor;
                    break;
                }
            }
        } while (node != endingNode);
        criticalPath.add(endingNode);
        return criticalPath;
    }

    public Map<Node, Double> getEarlyStartTimes() {
        return new LinkedHashMap<>(earlyStartTimes);
    }

    public Map<Node, Double> getEarlyFinishTimes() {
        return new LinkedHashMap<>(earlyFinishTimes);
    }

    public Map<Node, Double> getLateStartTimes() {
        return new LinkedHashMap<>(lateStartTimes);
    }

    public Map<Node, Double> getLateFinishTimes() {
        return new LinkedHashMap<>(lateFinishTimes);
    }

    public double getFinishTime() {
        return finishTime;
    }

    public Map<Node, Double> getSlack() {
        return new LinkedHashMap<>(slack);
    }

}
```

APPENDIX F. SimulationAnalyzeBase.java

```
package htn.test;

import ds.maker.RefIdHolder;
import edu.nps.moves.network.CPMComponent;
import edu.nps.moves.network.Node;
import edu.nps.moves.network.SimpleNode;
import edu.nps.moves.network.StartFinishListener;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.Date;
import java.util.Iterator;
import java.util.LinkedHashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import simkit.Schedule;
import simkit.random.RandomVariate;
import simkit.random.RandomVariateFactory;
import simkit.stat.SimpleStatsTally;
import trac.c2.cpm.CriticalPath;
import trac.c2.cpm.CycleException;
import trac.c2.cpm.example.network.SimplePrecedenceArc;
import trac.c2.cpm.example.network.SimpleTaskNetwork;
import trac.c2.cpm.example.network.SimpleTaskNode;
import trac.c2.cpm.network.PrecedenceArcIfc;

/**
 *
 * @author hasanbeker
 */
public class SimulationAnalyzeBase {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws FileNotFoundException, IOException, CycleException {
        String fileName = args.length == 0 ? "data/thesis.xlsx" : args[0];
        File inputFile = new File(fileName);
        System.out.println("Input file: " + inputFile.getAbsolutePath() + " " + inputFile.exists());

        InputStream inputStream = new FileInputStream(inputFile);
        Workbook workbook = null;
        if (fileName.endsWith(".xls")) {
            workbook = new HSSFWorkbook(inputStream);
        } else if (fileName.endsWith(".xlsx")) {
```

```
    workbook = new XSSFWorkbook(inputStream);
} else {
    throw new IllegalArgumentException("Wrong type of file: "
            + inputFile.getName());
}

SimpleTaskNetwork stn = new SimpleTaskNetwork();
stn.setName("Base");

List<Node> nodeList = new LinkedList<>();
List<SimpleTaskNode> allNodes = new ArrayList<>();
List<SimplePrecedenceArc> allArcs = new ArrayList<>();

Sheet sheet = workbook.getSheet("InputSheetBase");

boolean firstRowRead = false;
for (Iterator<Row> rowIter = sheet.rowIterator(); rowIter.hasNext();) {
    Row nextRow = rowIter.next();
    if (!firstRowRead) {
        firstRowRead = true;
        continue;
    }

    SimpleTaskNode nextNode = new SimpleTaskNode();
    nextNode.setName(nextRow.getCell(2).getStringCellValue());
    allNodes.add(nextNode);

    RefIdHolder.putRefId("TaskNode", nextNode.getName(), nextNode);

    RandomVariate activity = RandomVariateFactory.
            getInstance("Triangle", nextRow.getCell(4).getNumericCellValue(),
                    nextRow.getCell(6).getNumericCellValue(),
                    nextRow.getCell(5).getNumericCellValue());

    nextNode.addDurationData(activity);

    stn.addTaskNode(nextNode);

    SimpleNode simpleNode = new SimpleNode(nextRow.getCell(1).getStringCellValue(),
            nextRow.getCell(2).getStringCellValue(),
            nextRow.getCell(3).getStringCellValue(),
            activity);

    nodeList.add(simpleNode);
}

for (int rowNum = 1; rowNum < sheet.getLastRowNum(); ++rowNum) {
    Row nextRow = sheet.getRow(rowNum);

    Node preSimNodes = nodeList.get(rowNum - 1);
    SimpleTaskNode predecessor = allNodes.get(rowNum - 1);
    for (int cellNum = 7; cellNum < nextRow.getPhysicalNumberOfCells(); ++cellNum) {
        Cell cell = nextRow.getCell(cellNum);
        if (cell.getCellType() == Cell.CELL_TYPE_NUMERIC) {
            int successorIndex = new Double(cell.getNumericCellValue()).intValue();
            SimpleTaskNode successorNode = allNodes.get(successorIndex);
            SimplePrecedenceArc arc = new SimplePrecedenceArc();
            arc.setFromNode(predecessor.getName());
            arc.setToNode(successorNode.getName());
            arc.setName(predecessor + " -> " + successorNode);
            predecessor.addOutgoingArc(arc);
```

```
            successorNode.addIncomingArc(arc);
            allArcs.add(arc);
            stn.addPrecedenceArc(arc);

            Node sucSimNodes = nodeList.get(successorIndex);
            preSimNodes.addSuccessor(sucSimNodes);
          }

        }

      }

      CPMComponent cpmComponent = new CPMComponent(nodeList.get(0), nodeList.get(nodeList.size()
- 1));

      StartFinishListener startFinishListener = new StartFinishListener();
      cpmComponent.addSimEventListener(startFinishListener);

     Sheet sheet4 = workbook.getSheet("Deadline and Replication");
      double deadline = 0;
      double numberReplications =0;
      boolean firstRowRead1 = false;
      for (Iterator<Row> rowIter = sheet4.rowIterator(); rowIter.hasNext();) {
        Row nextRow1 = rowIter.next();
        if (!firstRowRead1) {
          firstRowRead1 = true;
          continue;
        }
       deadline =  nextRow1.getCell(0).getNumericCellValue();
       numberReplications = nextRow1.getCell(1).getNumericCellValue();

      }
      SimpleStatsTally deadlineMetStats = new SimpleStatsTally("Deadline met");


      if (workbook.getSheet("OutputSheetBase") != null) {
        workbook.removeSheetAt(workbook.getSheetIndex("OutputSheetBase"));
      }
      Sheet sheet2 = workbook.createSheet("OutputSheetBase");
      List<Object[]> data = new ArrayList<>();
      data.add(new Object[]{"Scenario", "Run", "Node", "Event", "Time"});

//        This will hold the replication critical paths keyed by replication number
     Map<Integer, List<Node>> replicationCPs = new LinkedHashMap<>();

      for (int rep = 1; rep <= numberReplications; ++rep) {
        System.out.println("======Replication #" + rep+"=======");
        //System.out.println("\n");
        List<Object> findCriticalPath = CriticalPath.findCriticalPath(stn);
        System.out.println("Critical Path for Replication"+" "+rep+" (based on mean): ");
        for (Object obj : findCriticalPath) {
          System.out.print(((SimplePrecedenceArc) obj).getName() + " , ");

        }
        System.out.println();

        Schedule.setVerbose(false);

        Schedule.reset();
        Schedule.startSimulation();
        deadlineMetStats.newObservation(cpmComponent.getFinishTime() <= deadline);
```

```
            List<Node> criticalPath =
                    startFinishListener.getCriticalPath(cpmComponent.getStartingNode(),
                            cpmComponent.getEndingNode());
//          Add the replication critical path
            replicationCPs.put(rep, criticalPath);
//          Print out the replication critical path to the console (for debugging purposes)
            StringBuilder criticalPathStr = new StringBuilder();
            for (Node n : criticalPath) {
                criticalPathStr.append(n.getTaskName());
                 criticalPathStr.append(" -> ");
            }
            criticalPathStr.delete(criticalPathStr.length() - 4, criticalPathStr.length());
            System.out.println("Critical Path for Replication"+" "+rep+" (based on generated times): ");
            System.out.println(criticalPathStr);


            Map<Node, Double> startTimes = startFinishListener.getEarlyStartTimes();
            Map<Node, Double> finishTimes = startFinishListener.getEarlyFinishTimes();


            for (Node node : startTimes.keySet()) {
                double start = startTimes.get(node);
                double finish = finishTimes.get(node);

                data.add(new Object[]{stn.getName(), rep, node.getTaskName(), "Start", start});
                data.add(new Object[]{stn.getName(), rep, node.getTaskName(), "Finish", finish});

            }

            for (int rownum = 0; rownum < data.size(); ++rownum) {

                Row row = sheet2.createRow(rownum);
                Object[] objArr = data.get(rownum);
                int cellnum = 0;
                for (Object obj : objArr) {
                    Cell cell = row.createCell(cellnum++);

                    if (obj instanceof Date) {
                        cell.setCellValue((Date) obj);
                    } else if (obj instanceof Boolean) {
                        cell.setCellValue((Boolean) obj);
                    } else if (obj instanceof String) {
                        cell.setCellValue((String) obj);
                    } else if (obj instanceof Double || obj instanceof Integer) {
                        cell.setCellValue(((Number) obj).doubleValue());
                    }
                }

            }
            System.out.println("Finish Time for Replication"+" "+rep+": "+cpmComponent.getFinishTime());
        }
         if (workbook.getSheet("BaseScenarioStat") != null) {
            workbook.removeSheetAt(workbook.getSheetIndex("BaseScenarioStat"));
        }
        Sheet sheet3 = workbook.createSheet("BaseScenarioStat");
        List<Object[]> data2 = new ArrayList<>();
        data2.add(new Object[]{"Deadline", "Deadline Met Percantage"});
        data2.add(new Object[]{deadline, deadlineMetStats.getMean()*100+"%"});
        data2.add(new Object[]{"Critical Paths for Base Scenario"});
        List<Object> criticalPath = CriticalPath.findCriticalPath(stn);
```

```
        for (int rownum = 0; rownum < data2.size(); ++rownum) {
            Row row2 = sheet3.createRow(rownum);
            Object[] objArr2 = data2.get(rownum);
            int cellnum = 0;
            for (Object obj : objArr2) {
                Cell cell2 = row2.createCell(cellnum++);

                if (obj instanceof Date) {
                    cell2.setCellValue((Date) obj);
                } else if (obj instanceof Boolean) {
                    cell2.setCellValue((Boolean) obj);
                } else if (obj instanceof String) {
                    cell2.setCellValue((String) obj);
                } else if (obj instanceof Double || obj instanceof Integer) {
                    cell2.setCellValue(((Number) obj).doubleValue());
                }
            }
        }
//      First write the critical path based on the mean
        Row meanRow = sheet3.createRow(data2.size());
        meanRow.createCell(0).setCellValue("mean");
        for (int i = 0; i < criticalPath.size(); ++i) {
            PrecedenceArcIfc arc = (PrecedenceArcIfc) criticalPath.get(i);
            SimpleTaskNode fromNode = (SimpleTaskNode) arc.getFromNode();
            meanRow.createCell(i + 1).setCellValue(fromNode.getName());
            System.out.print(fromNode.getName() + " -> ");
        }
        PrecedenceArcIfc lastArcOnCP = (PrecedenceArcIfc) criticalPath.get(criticalPath.size() - 1);
        SimpleTaskNode lastNode = (SimpleTaskNode) lastArcOnCP.getToNode();
        meanRow.createCell(criticalPath.size() + 1).setCellValue(lastNode.getName());
        System.out.println(lastNode.getName());

//      Now write the critical path for each replication, one per row
        for (int rep : replicationCPs.keySet()) {
            Row repRow = sheet3.createRow(data2.size() + rep);
            int column = 0;
            repRow.createCell(column).setCellValue(rep);
            for (Node node : replicationCPs.get(rep)) {
                column += 1;
                Cell cell = repRow.createCell(column);
                cell.setCellValue(node.getTaskName());
            }
        }
        System.out.printf("Prob{deadline of %.3f met} = %.3f%n", deadline, deadlineMetStats.getMean());
        try {
            File outputFile = new File(args.length == 0 ? "data/thesis.xlsx" : args[0]);
            FileOutputStream out = new FileOutputStream(outputFile);
            workbook.write(out);
            out.close();
            System.out.println("Excel written successfully to"
                    + outputFile.getAbsolutePath());

        } catch (FileNotFoundException e) {
        } catch (IOException e) {
        }

        workbook.close();
    }

}
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX G. BaseScenariStat Sheet

Critical Paths for Base Scenario

mean	Start	A	C1	C2	D4	D5	D6	F1	F2	F3	F4	I	Finish
1	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
2	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
3	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
4	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
5	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
6	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
7	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
8	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
9	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
10	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
11	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
12	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
13	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
14	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
15	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
16	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
17	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
18	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
19	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
20	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
21	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
22	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
23	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
24	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
25	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
26	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
27	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
28	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
29	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
30	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
31	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
32	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
33	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
34	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
35	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
36	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
37	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish

38	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
39	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
40	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
41	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
42	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
43	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
44	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
45	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
46	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
47	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
48	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
49	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
50	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
51	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
52	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
53	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
54	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
55	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
56	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
57	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
58	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
59	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
60	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
61	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
62	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
63	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
64	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
65	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
66	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
67	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
68	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
69	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
70	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
71	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
72	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
73	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
74	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
75	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
76	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
77	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
78	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
79	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish

80	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
81	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
82	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
83	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
84	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
85	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
86	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
87	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
88	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
89	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
90	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
91	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
92	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
93	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
94	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
95	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
96	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
97	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
98	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
99	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
100	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX H. TechScenarioStat Sheet

Critical Paths for Tech Scenario

mean	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
1	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
2	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
3	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
4	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
5	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
6	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
7	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
8	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
9	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
10	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
11	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
12	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
13	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
14	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
15	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
16	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
17	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
18	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
19	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
20	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
21	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
22	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
23	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
24	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
25	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
26	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
27	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
28	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
29	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
30	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
31	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
32	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
33	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
34	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
35	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
36	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
37	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	

38	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
39	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
40	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
41	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
42	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
43	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
44	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
45	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
46	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
47	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
48	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
49	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
50	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
51	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
52	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
53	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
54	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
55	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
56	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
57	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
58	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
59	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
60	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
61	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
62	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
63	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
64	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
65	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
66	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
67	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
68	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
69	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
70	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
71	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
72	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
73	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
74	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
75	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
76	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
77	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
78	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
79	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	

80	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
81	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
82	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
83	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
84	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
85	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
86	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
87	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
88	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
89	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
90	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
91	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
92	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
93	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
94	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
95	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
96	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
97	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
98	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
99	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
100	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX I. MCScenarioStat Sheet

Critical Paths for MisCom Scenario

mean	Start	A	C1	C2	D4	D5	D6	F1	F2	F3	F4	I	Finish
1	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
2	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
3	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
4	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
5	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
6	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
7	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
8	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
9	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
10	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
11	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
12	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
13	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
14	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
15	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
16	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
17	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
18	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
19	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
20	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
21	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
22	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
23	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
24	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
25	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
26	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
27	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
28	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
29	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
30	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
31	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
32	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
33	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
34	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
35	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
36	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
37	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish

38	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
39	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
40	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
41	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
42	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
43	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
44	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
45	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
46	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
47	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
48	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
49	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
50	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
51	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
52	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
53	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
54	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
55	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
56	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
57	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
58	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
59	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
60	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
61	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
62	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
63	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
64	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
65	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
66	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
67	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
68	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
69	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
70	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
71	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
72	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
73	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
74	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
75	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
76	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
77	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
78	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
79	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish

80	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
81	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
82	Start	A	B1	B2	D1	D2	D3	D4	D5	D6	I	Finish	
83	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
84	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
85	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
86	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
87	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
88	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
89	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
90	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
91	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
92	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
93	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
94	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
95	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
96	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
97	Start	A	C1	E1	E2	E3	E4	D4	D5	D6	I	Finish	
98	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
99	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish
100	Start	A	C1	C2	F1	F2	F3	F4	D4	D5	D6	I	Finish

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Ben-Shalom, U., & Shamir, E. (2011). Mission command between theory and practice: The case of the IDF. *Defense & Security Analysis, 27*(2), 101.

Blitzkrieg. (n.d.). *Wikipedia*. Retrieved January 27, 2015, from http://en.wikipedia.org/wiki/Blitzkrieg

Buss, A. H. (1996). Modeling with event graphs. In *Proceedings of the 1996 Winter Simulation Conference* (pp. 153–160). doi:10.1109/WSC.1996.873273

Buss, A. (2011). *MV3302 discrete event simulation modelling class notes.* Unpublished manuscript.

Clausewitz, C. V. (2007). *On war.* (M. Howard & P. Paret, Trans.). New York, NY: Oxford University Press.

Department of the Army. (2012a). *Mission command* (Army Doctrine Publication [ADP] 6-0). Washington, DC: Author.

Department of the Army. (2012b). *Mission command* (Army Doctrine Reference Publication [ADRP] 6-0). Washington, DC: Author.

Flynn, M., & Schrankel, C. (2013). Applying mission command through the operations process. *Military Review, 93*(2), 25.

Hattes, K. A. (1999). *Special operations mission planning and analysis support system* (Master's thesis). Monterey, CA: Naval Postgraduate School.

Hillier, F. S., & Lieberman, G. J. (1995). *Introduction to operation research*. New York, NY: McGraw-Hill.

Mind Tools Community. (2015). Critical path analysis and PERT charts. Retrieved February 18, 2015 from http://www.mindtools.com/critpath.html

Pryer, D. A. (2013). Growing leaders who practice mission command and win the peace. *Military Review, 93*(6), 31–41.

Sanchez, P. J. (2007). Fundamentals of simulation modeling. In *Proceedings of the 2007 Winter Simulation Conference* (pp. 54–62). doi:10.1109/WSC.2007.4419588

Shamir, E. (2011). *Transforming command: The pursuit of mission command in the U.S., British, and Israeli armies*. Stanford, CA: Stanford University Press.

Wiest, D. J., & Levy, K. F. (1969). *A management guide to PERT/CPM*. Englewood Cliffs, NJ: Prentice-Hall.

Triangular distribution. (n.d.). *Wikipedia*. Retrieved March 12, 2015, from http://en.wikipedia.org/wiki/Triangular_distribution

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California